Sincronización de Flujo Multimedia a través de Relaciones Causales.

Raúl Jacinto M. [1] Raúl García M. [2] Roberto Gómez C. [3] Felix Corchado [1]

Resumen:

El diseño de aplicaciones multimedia requiere la resolución de la sincronización en el espacio y en el tiempo de los objetos multimedia que lo componen. En particular, cuando se diseñan aplicaciones multimedia distribuidas se requiere la transmisión de objetos multimedia entre los participantes de la aplicación. En este trabajo se propone un mecanismo generale que permite el manejo sincronizado de los objetos multimedia (voz, texto, gráficas, imagen, sonido) para dar sentido a la aplicación final.

Palabras-clave: *Sistemas Distribuidos, Multimedia, Sincronización*

CINVESTAV-Guadalajara [1]
Ave. Lopez Mateos Sur 590, Guadalajara, México.
e-mail: rjacinto@gdl.cinvestav.mx
e-mail: framos@gdl.cinvestav.mx

♣

Inst. Investigaciones Eléctricas [2]
Av. Reforma 113, Col. Palma, 62490, Temixco Morelos, México
e-mail: rgarcia@iie.org.mx

♣

ITESM-CEM [3] **- Dpto. Ciencias Computacionales**
Apdo. Postal 2 Módulo de Servicio Postal,
Atizápan Zaragoza, 52296 Edo. de México, México
e-mail: rogomez@campus.cem.itesm.mx

1 Introducción

El manejo de aplicaciones multimedia distribuidas requiere el considerar los aspectos de sincronización temporal y espacial relacionadas con el procesamiento, transporte, almacenamiento, recuperación y presentación de datos, sonido, imágenes y video. El problema de la sincronización multimedia presenta dos aspectos principales:

- la sincronización intra-objetos. Este aspecto involucra el respeto a la dependencia temporal de los elementos que conforman al objeto. En términos de tráfico en una red, es equivalente al ensamblaje correcto de los paquetes concernientes al objeto.
- La sincronización inter-objetos. Este aspecto involucra la sincronización entre los objetos multimedia de la aplicación. En términos de tráfico, es equivalente al manejo de objetos ligados temporalmente, por ejemplo la voz y la imagen en un video.

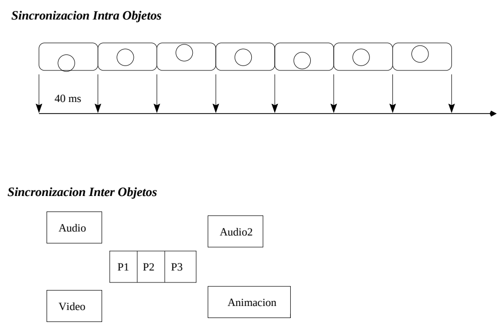

Figura 1: *La sincronización intra e interobjetos*

Una aplicación multimedia típica es una video-conferencia, la cual está compuesta de (al menos) objetos de audio y objetos de video con anchos de banda y requerimientos temporales distintos. En este tipo de aplicaciones se requiere la determinación dinámica de las signaturas temporales de cada objeto producido por la aplicación. En el caso de aplicaciones estáticas o precompiladas, tales como visitas virtuales a un museo o el uso de un sistema tutorial, una vez la aplicación definida las relaciones entre los diferentes objetos no cambian.

El trabajo necesario para realizar la sincronización de estas fuentes es complejo e involucra la consideración de las relaciones de dependencia y causalidad entre los objetos multimedia de la aplicación. Para tratar con este tipo de problemas, numerosos autores han propuesto técnicas como el estampado de mensajes con el instante de creación del objeto o la adición de información lógica que condiciona la entrega del objeto generado.

El segundo enfoque es más general, en cuanto a que considera, por cada objeto, relaciones de dependencia y causalidad entre inter e intra objetos al mismo tiempo que sus signaturas temporales. Este enfoque requiere de una nomenclatura pesada, al menos a nuestro parecer, y de un cómputo adicional en el receptor para tratar la infomación y entregarla sincronizada al usuario.

Particularmente en el caso de aplicaciones precompiladas resulta, por razones prácticas, conveniente el que el servidor donde radica la información calcule las dependencias, de tal forma que se libere al cliente al máximo posible de ejecutar trabajo innecesario y de requerir software específico. En efecto, es mejor realizar una vez el trabajo en el servidor que ejecutarlo cada vez que un cliente lo requiera. En el caso de aplicaciones dinámicas, tipo video conferencia, ambos lados de la aplicación están simétricamente cargados por lo que no hay razones que sugieran que transferir el trabajo de un procesador al otro sea la mejor política.

En este trabajo, proponemos un formalismo que permite la expresión de las dependencias temporales de manera simple y expresiva. En base a este formalismo se propone un mecanismo general de sincronización para tratar con los dos aspectos principales del problema. Este trabajo se basa en reportes previos de otros investigadores [1][3][2] los cuales, sin embargo, presentan un formalismo que resulta pesado de manipular, o bien sólo se enfocan a uno de los dos aspectos del problema. Dentro de este trabajo se presenta un análisis del tratamiento de la sincronización en caso de pérdidas de la información. Este es un problema importante en cuanto a que la pérdida de la información puede ser detectada erronéamente; un tratamiento adecuado es, por lo tanto, básico para la sincronización de la información.

En la sección dos se presenta un ejemplo de aplicación multimedia típica, en la sección tres se proporciona un resumen de la terminología y conceptos utilizados en este artículo, la sección cuatro presenta los segmentos de la teoría de intervalos utilizados en este trabajo y su aplicación en sincronización de flujo multimedia, la sección cinco presenta el núcleo de este trabajo y finalmente en la sección seis se presentan las conclusiones y algunos trabajos futuros.

2 Ejemplo de aplicación multimedia

El ejemplo de aplicación multimedia se basa en una aplicación de educación interactiva a distancia. El ejemplo está tomado de [1]. La aplicación se considera distribuida en tres nodos:

- Un servidor síncrono SS,
- Un servidor asíncrono AS,
- Una terminal del estudiante E

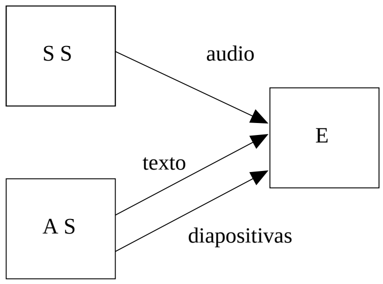

Figura 2: *Esquema general del ejemplo*

El SS provee al estudiante con información audible, mientras que el AS proporciona al estudiante información en texto y en diapositivas. El tipo de transferencia multimedia a E se caracteriza como proveniente de distintas fuentes a un sólo destino.

La parte de la aplicación de educación considerada aqui se componene de las tres fases siguientes : una introducción, los servidores mandan varias unidades de información al estudiante para presentar un tópico particular, una pregunta, los servidores mandan diferentes unidades de información al estudiante para cuestionarlo sobre un punto en particular, y finalmente una fase de evaluación, los servidores mandan al usuario diferentes unidades de infomación para comentar sus respuestas. La figura 3 muestra el escenario de alto nivel relacionado a la entrega del haz de información, incluyendo la duración de la presentación de cada unidad de información enviada por los servidores. Finalmente se asume que el estudiante responde a las preguntas por medio de una conexión de control entre él y SS.

Considere los requerimientos de sincronización que pueden asociarse con las diferentes fases de la aplicación. En la introducción, varias unidades de información se transfieren de los servidores al estudiante, i.e. una unidad de diapositivas, cuatro unidades de audio (cada unidad de audio corresponde a algún segmento de audio codificado) y dos unidades de información de texto. En

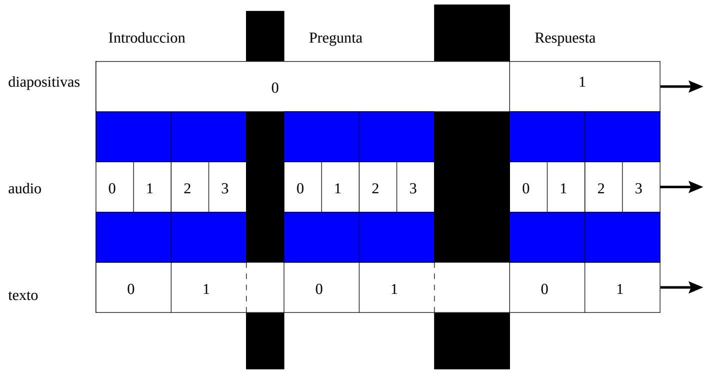

Figura 3: *Los diferentes mensajes enviados durante la aplicación*

esta fase, existe una restricción fuerte de entregar las unidades de audio y diapositivas. En otras palabras, la entrega de un sólo tipo de información (audio o diapositiva) es inútil desde el punto de vista de la aplicación[1]. Una restricción tan fuerte no es necesaria para las unidades de texto, se asume que su fase de aplicación puede proceder aún si el texto no ha sido presentado enteramente al estudiante. La fase de pregunta incluye la presentación continua de las diapositivas previas, así como la de las cuatro nuevas unidades de audio y de las dos nuevas unidades de texto. En esta fase el requerimiento mayor es sobre las unidades de texto, ya que estas describen las diferentes opciones disponibles al estudiante, durante las preguntas efectuadas a tráves del uso del audio.

Finalmente, la fase de evaluación incluye la presentación de una diapositiva, cuatro unidades de audio y una información adicional de texto. La información trasferida al usuario depende de la respuesta del estudiante. En esta fase, la entrega de información de audio no es obligatoria, ya que se asume que el estudiante estará suficientemente consciente de la validez de sus respuestas gracias a las otras unidades de informacion entregadas a él (texto o diapositivas).

Las tres fases de la aplicación son necesarias, en consecuencia si una de las fases no se termina correctamente (por ejemplo debido a que las unidades de información necesarias no estaban presentes en el momento adecuado) entonces ninguna de las fases subsecuentes ocurre.

2.1 Expresión de las Relaciones de Causalidad.

La aplicación descrita anteriormente puede describirse en base al grafo de dependencias de la figura 4. Las relaciones de causalidad se expresan por una flecha dirigida entre dos nodos que representan unidades de información, una flecha de a a b significa que a depende de b. Un circuito en este grafo caracteriza la presencia de un punto de sincronización entre las respectivas unidades de información.

Con respecto a la aplicación ejemplo, es posible darse cuenta que la relación, en la fase de introducción, entre la primera información de audio ai_0 y la pimera unidad de texto ti_0 implica que si ai_0 no puede ser entregada entonces ti_0 no puede ser entregada tampoco. En otras palabras, desde el punto de vista de la aplicación es inútil entregar ti_0 si ai_0 no puede entregarse. En la misma fase de introducción, la dependencia acoplada de las unidades de información ai_0 y si_0 implica que la entrega de una requiere de la entrega de la otra. Este tipo de dependencia caracteriza la sincronización, lo cual resulta en la entrega simultánea de las unidades de información involucradas en la dependencia cruzada.

Las relaciones causales entre dos unidades consecutivas de información significa que estas unidades no pueden perderse. En la fase de evaluación, las unidades de información aw_1, aw_2, aw_3 sólo dependen de aw_0 , esta dependencia se requiere para iniciar la fase de evaluación de respuestas. Del

[1] Un ejemplo de esto sería la visión de un monumento histórico sin ninguna explicación, el estudiante díficilmente podría ubicar el monumento y su importancia sólo a partir de una diapositiva .

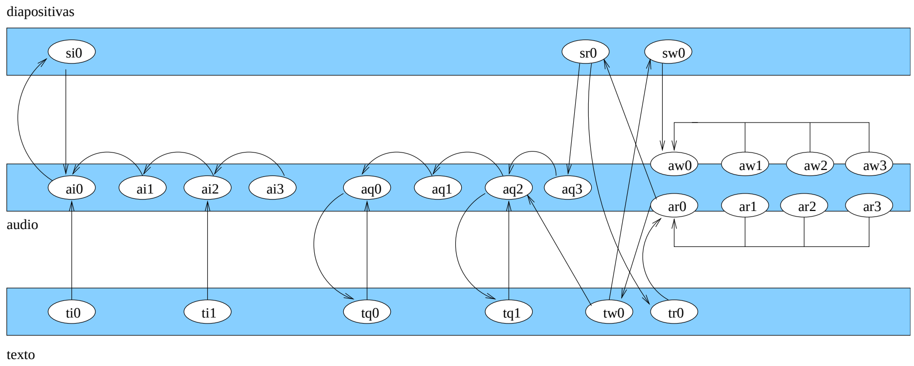

Figura 4: *Las relaciones causalidad de la aplicación*

grafo de la aplicación puede verse que los puntos de sincronización $\{aq_0,\ tq_0\}$, $\{aq_1,\ tq_1\}$ forzan la presentación simultánea de las unidades de audio y texto durante la fase de preguntas.

Una notación conveniente para expresar estas relaciones es la siguiente $ui =_d expr$ que implica que la unidad de información asociada al identificador ui depende de la expresión *expr*. En donde *expr* es una expresión lógica en forma disjuntiva de un número finito de variables booleanas, las variables booleanas son identificadores de unidades de información. Dependiendo de si estas están prefijadas o no por el operador lógico *no* (denotado por $\not{}$) las variables caracterizan las premisas negativas o positivas definidas dentro de una conjunción particular de una expresion de dependencias.

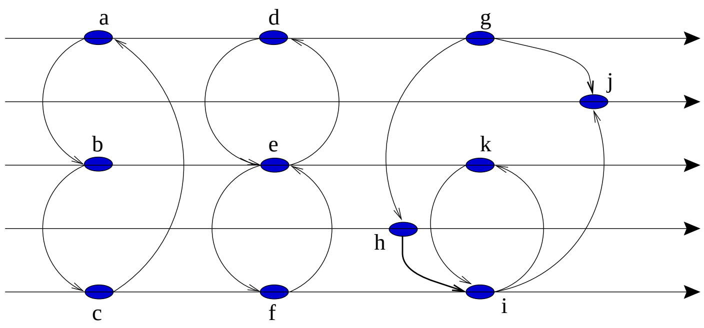

Figura 5: *Ejemplo dependencias causalidad*

Para ilustrar las dependencias, considere el ejemplo de la figura 5:

- El punto simple de sincronización de las unidades $\{a, b, c\}$ donde $a =_d b, b =_d c$ y $c =_d a$. Las dependencias definen un punto de sincronización simple, ya que existe un ciclo en las dependencias.
- Las configuraciones $\{d, e\}$ y $\{e, f\}$ donde $d =_d e$, $e =_d d \wedge f$ (la entrega de e depende de la de d y f). Estas configuraciones constituyen un punto sencillo de sincronización, y pueden ser tratadas como tales.
- La configuración $\{g, h, i, j\}$ y $\{i, k\}$ donde $g =_d h$, $h =_d i$, $i =_d k \vee j$, $j =_d g$, $k =_d i$, puede dar lugar a dos puntos distintos de sincronización dependiendo de las signaturas temporales de las configuraciones.
- La configuración $\{l, m\}$ con $l =_d m$, que expresa la entrega de la unidad l sólo en caso de no ser capaz de entregar m. En este caso la signatura temporal de l debe caer dentro del complemento

de la signatura temporal de m. Más aún, la signatura temporal de l, sólo puede estar en el intervalo derecho de la signatura de m.

La entrega efectiva de las unidades de información del flujo multimedia depende de las restricciones de entrega impuestas por las dependencias y por las restricciones temporales debidas a la naturaleza del flujo mismo. Sea $\Delta m = [\underline{t}^m, \bar{t}^m]$, con $\underline{t}$ el instante mínimo de entrega y $\bar{t}$ el instante máximo, el llamado intervalo de tiempo de entrega asociado a la unidad de información m. Este intervalo define la cantidad máxima de tiempo ($\delta = \underline{t}^m - \bar{t}^m$) durante la cual m puede almacenarse dentro de la entidad receptora antes de su entrega (se asume que una vez que la unidad se entrega, su consumición es instántanea) . Básicamente este intervalo puede definirse en dos maneras, dependiendo de la clase de requerimientos temporales asociados al flujo:

1. Si no existe un mecanismo de compensación del jitter[2] para aplicárselo al flujo, entonces δ puede definirse como un parámetro de la Calidad de Servicio (QoS) negociada para el mecanismo de entrega condicional.
2. De otra manera, δ puede derivarse del jitter residual permitido por el mecanismo de compensación.

La diferencia básica entre ambos casos radica en el instante en que Δm inicia. De acuerdo a (1), Δm inicia en el instante tr^m en que la unidad se recibe por la unidad de transporte y está disponible para su entrega. i.e. $\Delta m = [tr^m, tr^m + \delta]$. Si, el caso (2) aplica, entonces puede ser que :

- La unidad esté disponible antes de su instante inicial de entrega, $tr^m \leq \underline{t}^m$ y $\Delta m = [\underline{t}^m, \bar{t}^m]$.
- que llegue durante su intervalo de entrega $\underline{t}^m \leq tr^m \leq \bar{t}^m$y $\Delta m = [tr^m \ , \bar{t}^m]$.
- o que llegue tarde, en cuyo caso m no puede entregarse debido a una recepción tardía y debe ser descartada. No existe un intervalo de entrega asociado.

2.2 Arquitectura de la aplicación.

Los mecanismos propuestos se ubican sobre la capa de transporte, típicamente TCP-IP que garantiza un servicio básico orientado conexión sin acuse de recepción, con un ancho de banda adecuado (una hipotésis optimista), con una tasa de pérdidas de paquetes limitada a un porcentaje específico y posiblemente con mecanismos de compensación del jitter para flujos isócronos. La arquitectura propuesta se ilustra en la figura 6, los mecanismos propuestos proporcionan el servicio de sincronización a la capa aplicación, del lado emisor el envío etiquetado del flujo multimedia, mientras que del lado emisor la recepción sincronizada del mismo flujo. Obviamente se asume un tiempo de retardo aceptable en el transporte de la información entre los dos extremos.

A cada unidad de información se le asocia una expresión de dependencias que condiciona lógicamente la entrega de la unidad. Igualmente las unidades de información tienen asociada su signatura temporal Δm. Esta signatura temporal está condicionada por las dependencias de la unidad, la manera en que las dependencias afectan esta signatura se especifica en la sección 4.

3 Teoría de Intervalos y su Aplicación en Sincronización Multimedia

3.1 Números Intervalos.

Un intervalo se define como un conjunto cerrado limitado de numeros reales $[a, b] = \{x : a \leq x \leq b\}$. De la misma manera que es posible representar números racionales a/b, números complejos $x + iy$ por un par ordenado de números, es posible representar un intervalo por una pareja ordenada , sea

[2]El jitter se define como el lapso permitido de desviación al punto exacto en que la acción debió producirse, proviene del inglés **j**ust **i**n **t**ime.

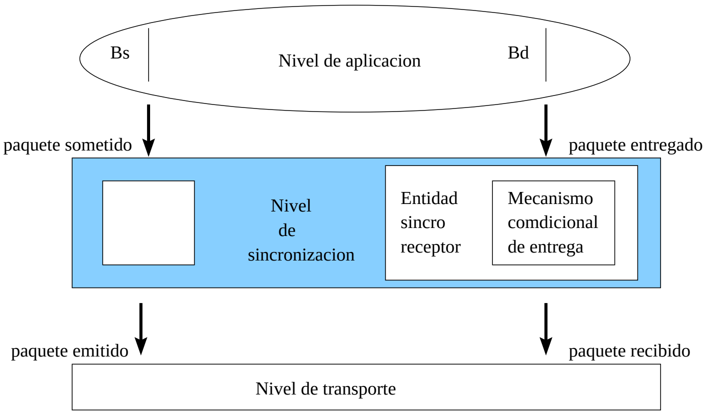

Figura 6: *La arquitecura de la aplicación*

X un intervalo, entonces $X = [\underline{X}, \overline{X}]$ en donde $\underline{X}$ es el límite mínimo y $\overline{X}$ es el límite máximo del intervalo. Los intervalos entonces permiten representar un conjunto de números reales por una nueva clase de números, llamados intervalos [7]

Definición 3.1 *Dos intervalos X, Y son iguales si sus correspondientes límites son iguales: $X = Y$ si y solo si $\overline{X} = \overline{Y}$ y $\underline{X}$=$\underline{Y}$.*

Definición 3.2 *Decimos que la intersección de dos intervalos es vacía si no existen elementos en común $X \bigcap Y = \emptyset$ si $\underline{X} < \overline{Y}$ o $\underline{Y} > \overline{X}$. Si la intersección es no vacía el resultado es a su vez un nuevo intervalo definido por*
$X \frown Y = [\max(\underline{X}, \underline{Y}), \min(\overline{X}, \overline{Y})]$.

Definición 3.3 *La unión de dos intervalos se define como $X \smile Y = [\min(\underline{X}, \underline{Y}), \max(\overline{X}, \overline{Y})]$ si la interseccion de X y Y es no vacía. En caso de que la intersección sea vacía el resultado es un vector de intervalos dado por $X \smile Y = (X, Y)$.*

Definición 3.4 *La relación de transitividad $X < Y$ se satisface si y sólo si $\overline{X} < \underline{Y}$.*

Definición 3.5 *Un intervalo está incluido en otro $X \subseteq Y$ si y sólo si $\underline{Y} \leq \underline{X}$ y $\overline{X} \leq \overline{Y}$.*

Definición 3.6 *El ancho (width) de un intervalo se define como $w(X) = \overline{|X} - \underline{X|}$*

Definición 3.7 *El complemento modificado de un intervalo se define como $\neg X = \{x : x \notin, x > \underline{X}\}$.*
Esta definición permite considerar el caso estricto en el que un objeto depende de la no ocurrencia del otro.

3.2 Los números intervalos y el tráfico multimedia.

La idea central de este trabajo es la conceptualización de un haz de flujos multimedia por medio de una matriz, en donde un renglón representa un flujo de objetos multimedia de la misma clase, mientras que las columnas representan los intervalos de entrega de cada objeto. Al conjunto de flujos se le conoce como un haz multimedia.

En esta representación los objetos que comparten la misma columna mantienen una relación de sincronía, mientras que los objetos en el mismo renglón tienen una relación de causalidad. Es posible etiquetar a cada objeto con expresiones lógicas que permitan expresar relaciones más complejas, estas relaciones sólo pueden ser de dependencia y no de sincronización.

Las relaciones entre los objetos consideradas son todas aquellas que puedan expresarse por medio de la lógica proposicional. Sin embargo es claro que los intervalos de entrega asociados a cada objeto son condicionados por estas relaciones, es posible entonces considerar dos tipos de relaciones:

- Sincronía: una relación de sincronía existe si la existencia temporal de dos objetos debe preservarse a toda costa.
- Dependencia: una relación de dependencia existe si la entrega de un objeto depende de la entrega de otros.

La separación de las relaciones en estos dos tipos facilita enormemente el manejo del haz multimedia. En base a esta separación, analizamos las relaciones de sincronía. Considere el caso de un punto de sincronía entre varios objetos.

Definición 3.8 *Sincronía fuerte: sea i el renglón y j la columna de la matriz M que representa un haz multimedia. $\Delta x_{i,j}$ representa los intervalos de los elementos del renglón i y de la columna j, la sincronía fuerte $sinc(\Delta x_{i,j}) = \frown_{i=1,n} \Delta x_{ij}$ en dónde n es el número de flujos en el haz.*

Expresado en términos de la aplicación la sincronía fuerte se define como la intersección de todos los intervalos de entrega de los objetos involucrados en la sincronización. Un análisis del problema revela que es necesario una sincronización más flexible, en el cual la sincronización todavía se considera posible si los objetos se entregan dentro del mayor intervalo de entrega de los objetos involucrados.

Definición 3.9 *Sincronía débil: sea i, el renglón y j la columna de la matriz M que representa un haz multimedia. $\Delta x_{i,j}$ representa los intervalos de los elementos del renglón i y de la columna j, $dsinc(\Delta x_{i,j}) = [\min_{i=1,n}(x_{i,j}). \max_{i=1,n}(x_{ij})]$*

3.3 Relaciones de dependencia

Analicemos el caso de las relaciones de dependencia, para lo cual considere el caso siguiente (por comodidad regresamos a la notación de objetos por literales en lugar de elementos de la matriz) $d =_d \neg ab \ \vee \neg ba$ (d depende de manera exclusiva de a y b), en este caso existen potencialmente dos intervalos de entrega asociados a d ($\Delta d = (\Delta a \smile \Delta b) - (\Delta a \frown \Delta b)$), si atendemos a la sémantica tradicional del o exclusivo. En una aplicación del tipo tráfico multimedia, la sémantica asociada es tal que se requiere potencialmente la espera durante todo el intervalo ($\Delta a \smile \Delta b$) para determinar si alguno de los términos de la relación se cumple. Una discusión similar es posible para todas las relaciones tipo o ($\vee$).

Un examen atento muestra sin embargo que analizando por separado cada término de la disjunción, los intervalos se definen como $\Delta a \frown \Delta \neg b$ con el intervalo correspondiente $\Delta \neg a \frown \Delta b$. En este caso dos posibilidades están presentes:

- $a < b$, en cuyo caso la intersección es nula y la única posibilidad real es $\neg ab$ y por lo tanto hay que esperar potencialmente durante todo el intervalo de b para saber si d puede o no ser entregado.
- $a \geq b$ en cuyo caso la intersección es no nula y el único intervalo posible es el de alguno de los elementos de la disyunción.

El problema cuando se tienen varios términos en una expresión lógica es que potencialmente se requiere un intervalo de entrega por cada término. En este caso el intervalo es formado con el menor de los límites inferiores y con el mayor de los límites superiores, esta selección permite mantener uniforme la cantidad de información unida a cada objeto.

Con respecto a las conjunciones, la sémantica debe quedar claramente establecida, dos casos son posibles:

- un primer caso en el que la conjunción implica la entrega de a y b en un intervalo de ejecución de ambos
- un segundo caso dónde el intervalo se define a partir del punto en donde la entrega de ambos es posible, es decir a puede ser entregado primero y b entregado en cualquier momento de su intervalo, cumpliendo con el requerimiento de entrega de ambas unidades a y b.

Notaremos el primer caso de la manera tradicional (dos nombres unidos ab) con un intervalo de entrega dado por $\Delta a \frown \Delta b$ mientras que el segundo caso se denotará por $conj(a,b)$ con un intervalo de entrega dado por $\Delta a \frown \Delta b \smile \Delta b$ se asume que $\underline{a}<\underline{b}$ y $\overline{a}<\overline{b}$)

4 El Algoritmo Propuesto

El algoritmo propuesto se basa en la conceptualización de un tráfico multimedia por medio de una matriz de objetos $M(m,n)$, donde los elementos de la matriz son los objetos de un flujo multimedia. Cada renglón representa un flujo de objetos, las columnas representan posibilidades de sincronización, cada objeto tiene asociado un descriptor de dependencias y un intervalo de entrega. Inicialmente los intervalos de entrega se suponen dados por estimación del diseñador.

La idea es que del lado emisor existe un preproceso de la matriz para ajustar los intervalos de entrega de cada objeto. Este ajuste permite al lado receptor dedicarse exclusivamente a un trabajo de verificación de dependencias. Ambos procesos se basan en las definiciones dadas previamente.

El preproceso del lado emisor requiere entonces los siguientes pasos:

1. Ajustar los límites derechos (inferiores) de cada objeto. Reduciendo potencialmente el tiempo de entrega de cada objeto.
2. Ajustar los objetos en columnas con sincronía.
3. Ajustar las límites inferiores de cada objeto.

Una vez la matriz ajustada, cada vez que se requiera el envío de un objeto el emisor simplemente etiqueta al objeto con su expresión de dependencias y su intervalo de entrega. Si un elemento de una columna tiene en su expresión una función de sincronía $sinc()$, $dsinc()$ se considera que la sincronía es con todos los elementos de la columna.

Formalmente el ajuste de los límtes de los objetos se realiza de acuerdo al siguiente algoritmo

4.1 Descripción del algoritmo

Se asume que los relojes del cliente y servidores estan sincronizados, (para detalles de algoritmos de sincronía de relojes se puede ver [4, 5, 6]). Bajo esta hipótesis el cálculo de los tiempos de transmisión correspondiente a cada flujo es:

$$TTx = T_{local} - T_s(mensaje);$$

donde TT_x es el tiempo de transmisión total, T_{local} es la hora local en la cual se recibió el mensaje, y $T_s(mensaje)$ es la estampilla de tiempo del proceso emisor del mensaje, (recuerdese que los relojes estan sincronizados). Ahora bien, debido a las potenciales deferencias de velocidad, ajustar los tiempos de transmisión con cada recepción de segmentos de datos no es adecuado, por lo que se prefiere hacer el ajuste de acuerdo a la siguiente expresión:

$$TTx_{act} = (TTx_{old} + TTx_{new})/2;$$
$$TTx_{old} = TTx_{new}$$

La estimación del tiempo de transmisión (recuerde que en este trabajo sólo tratamos con los aspectos inter-objetos) permite la previsión del tiempo de almacenamiento de un objeto. Supongamos el caso de la figura 7 que se encuentra abajo:

con $[\underline{ai0}, \overline{ai0}]$ y un TTx_{si0} dado, sea:

$$t_{act} \ \in \ [\underline{ai0}, \overline{ai0}]\text{tal que} t_{act} + (TTx) > ai0$$

bajo estas circunstancias $ai0$ debe desecharse puesto que $si0$ no podrá ser recibido a tiempo, impidiendo con esto la entrega de $ti0$. El problema se deriva de la falta de información de $ai0$ acerca del

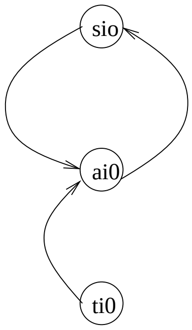

Figura 7: *Ejemplo de sincronia*

estado actual de $si0$, el cual puede estar en proceso de recepción al interior del objeto. Para evitar esta situación $ai0$ se mantiene en memoria ajustando el intervalo a:

$$\delta aio = [\underline{t_{act}}, \overline{a_{i0}} + t_{act}]$$

Del lado receptor se mantienen dos estructuras de datos una para almacenar los objetos no elegibles y otra para los objetos elegibles. Adicionalmente se mantiene una lista de objetos entregados. Una vez un objeto recibido de acuerdo al tiempo actual del emisor se determina si este es elegible o no para ubicarlo en su estructura de datos correspondiente, actualizando sus dependencias en caso necesario.

Un objeto elegible con todas sus dependencias satisfechas origina su entrega al usuario y simúltanemente la actualización de las dependencias de los objetos en ambas estructuras. Un objeto no elegible se transforma en elegible si su límte inferior es mayor que el La especificación del algoritmo se presenta en la figura 8.

$\forall e_i \in$ columna j **hacer**
    si $(e_i =_d expr)$ **y** $(t_{act} \in [\underline{e_i}, \overline{e_i}])$ **entonces**
        $\forall e'_i$ $(e_i \neq e'_i \in$ columna $j)$ **hacer**
            si (sinc(e_i, e'_i) **y** $(e'_i =_d expr)$ **entonces**
                entregar e_i, e'_i
                abortar
            si (dsinc(e_i, e'_i) **entonces**
                entregar e_i
                abortar
    fsi
    si $(e_i =_d expr)$ **y** $(t_{act} \geq [\underline{e_i}, \overline{e_i}]))$ **entonces**
        desechar e_i
    fsi
    $<$ actualiza dependencia en intervalo $>$
fin-hacer

Figura 8: *Especificación del algoritmo*

El enunciado *abortar* tiene por efecto el termino del algoritmo, los enunciados *entregar* y *desechar* representan el paso del objeto multimedia a la capa superior o el no paso de este.

5 Conclusiones y trabajo futuro

El trabajo presentado ataca el problema de la sincronización de tráfico multimedia interobjetos. La solución presentada, al basarse en el álgebra de intervalos, resulta de un bajo nivel de complejidad lo que facilita su implementación. Actualmente se trabaja en el desarrollo de un prototipo para validar la solución propuesta.

Dentro de las líneas de investigación futuras se considera: una evaluación adecuada del tráfico en la red para generar un ordenamiento predictivo de la entrega de los objetos multimedia. Una línea adicional de investigación es el estudio de la relación entre la sincronización inter e intra objetos, explotar esta relaciónpuede facilitar el manejo de aplicaciones que puedan soportar la entrega parcial de objetos. Por otro lado se piensa trabajar en el calculo del tiempo estimado de transmisión basandose en el trabajo [8], así como el tamaño de los buffers para el alamcenamiento de los objetos multimedia.

Referencias

[1] J. P. Courtiat, L.F. Rust da Costa C, R. Cruz de Oliveira "A general-Purpose Multimedia Syncronizarion Mechanism Based on Causal Relations" *IEEE Journals On Selected Areas in Commun., Vol 14, No. 1. pp 185-195 january. 1996*

[2] T.D.C. Little and A. Ghafoor "Syncronization and Storage Models for Multimedia Objects" *IEEE Journals On Selected Areas in Commun., Vol 8, No. 3. pp 413-427 apr. 1990*

[3] L. Daraine "Techniques de synchronisation pour les communications dans les systèmes haut débit multimedia" Ph. D. Dissertation Université Pierre et Marie Curie Paris, 1994.

[4] Lamport "Time, clocks and ordering of Events in a Distributed System", Comm. of the ACM, vol. 21, No.7, Julio 1978, pp. 558-565

[5] Flaviu Cristian, Probabilistic Clock Synchronization. Distributed Computing Vol. 3, pp. 146-158, 1989

[6] Gusella, R. and Zatti, "The Accuracy of the Clock Synchronization Achieved by TEMPO in Berkeley UNIX 4.3BSD", IEEE Trans. on Software Engineering, vol. 15, pp. 847-853, Jul. 1989

[7] Moore Ramon, "Methods and Applications of Interval Abalysis" SIAM Studies In Applied Mathemathics, 1979, (ISBN: 0-89871-161-1)

[8] Gibbon John, Little Thomas, "The Use of Network Delay Estimation for Multimedia Data Retrieval" IEEE Journal on selected areas in Communications, Vol. 14, No. 7, Sep. 1996