Um Mecanismo de Sincronização Intermídia com Taxa Mínima de Realimentação[1]

Marcelo Gonçalves Rubinstein
rubi@gta.ufrj.br

Otto Carlos Muniz Bandeira Duarte
otto@gta.ufrj.br

Grupo de Teleinformática e Automação - GTA
Universidade Federal do Rio de Janeiro
COPPE/EE - Programa de Engenharia Elétrica/Departamento de Eletrônica
Caixa Postal 68504 - CEP 21945-970 - Rio de Janeiro - RJ - Brasil

Resumo

De modo a sincronizar pares de mídias (mestra, escrava) de um feixe de sincronização, um servidor de sincronização utiliza unidades de realimentação para estimar os instantes de exibição das unidades de mídia. Este trabalho propõe uma extensão ao mecanismo de sincronização implementado no Grupo de Teleinformática e Automação (GTA) da UFRJ, de modo a diminuir o *overhead* adicional devido à transmissão de uma unidade de realimentação para cada unidade de mídia exibida na escrava. Dada a assincronia máxima permissível entre os pares de mídias, o servidor determina a menor taxa de transmissão pela escrava de unidades de realimentação. Para valores utilizados no protótipo do sistema de sincronização implementado, esta técnica possibilita uma taxa de envio de unidades de realimentação sete vezes menor, diminuindo substancialmente o processamento no servidor e o *overhead* na rede.

Abstract

In order to synchronize pairs of media (master, slave) of a synchronization bundle, a synchronization server uses feedback units to estimate playback times of media units. This work proposes an extension to the synchronization mechanism implemented in the Grupo de Teleinformática e Automação (GTA) of the UFRJ, in order to decrease the additional overhead due to the transmission of a feedback unit per every media unit played in the slave. Given the maximum tolerable asynchrony between the pairs of media, the server determines the minimum feedback transmission rate from the slave. For values used in the prototype of the implemented synchronization system, this technique enables a feedback transmission rate seven times smaller, essentially decreasing server processing and overhead on the network.

1 Introdução

Com o surgimento de novas aplicações envolvendo a integração de várias mídias, vários requisitos de Qualidade de Serviço (QoS) foram introduzidos, tais como o sincronismo entre as amostras de áudio e entre os quadros de vídeo. A integração das mídias requer técnicas de sincronização que devem reproduzir, no(s) destino(s), as relações temporais existentes na(s) fonte(s) quando da captura da(s) mídia(s). A apresentação de fluxos sincronizados é essencial para obter-se uma impressão de naturalidade, pois fluxos fora de sincronismo são percebidos como sendo algo artificial, estranho ou até mesmo enjoado [Stei96].

A sincronização pode ser classificada em relação ao fluxo envolvido e à forma de expressar as relações temporais. Em relação ao fluxo envolvido, são definidos dois tipos de sincronização: a intramídia (intrafluxo) e a intermídia (interfluxo). A **sincronização intramídia**, também chamada continuidade, corresponde à relação temporal existente entre

[1]Este trabalho foi realizado com recursos da UFRJ, FUJB, PROTEM-CC, CNPq, FAPERJ, CAPES e COFECUB.

um dado e o próximo dado de um mesmo fluxo, sendo este fluxo isócrono ou não. Para um fluxo isócrono, o intervalo entre a apresentação de dois dados seqüenciais deve ser fixo. A **sincronização intermídia** corresponde à relação temporal entre dados, de dois ou mais fluxos, isócronos ou não. Este tipo de sincronização se aplica também ao caso de duas ou mais instâncias de um mesmo tipo de mídia; por exemplo, um áudio sincronizado a outro áudio. Em relação à forma de expressar as relações temporais, a sincronização pode ser contínua ou sintética. A **sincronização contínua**, também chamada sincronização natural ou sincronização ao vivo, mantém relações temporais implícitas de modo contínuo, para uma ou mais mídias. Estas relações são expressas através da relação temporal *igual*. Se o objetivo da apresentação for exibir as mídias de maneira igual a que foram capturadas, esta sincronização deve ser utilizada. A **sincronização sintética**, também chamada sincronização baseada em eventos, serve-se de relações temporais explícitas entre objetos multimídias. Logo, as relações temporais entre estes objetos são especificadas artificialmente e podem ser dos tipos *antes*, *depois*, *durante* e outras, num total de treze [LiGh90]. Os dados envolvidos nas aplicações que se servem dos mecanismos de sincronização sintética são chamados dados pré-orquestrados [LiDu96].

Nos últimos anos, várias técnicas de sincronização foram propostas. O modelo de Little e Ghaffor [LiGh91] utiliza Redes de Petri para a Composição de Objetos (OCPNs) a fim de especificar as relações temporais entre os objetos multimídias. Um extrator de dependências condicionais é proposto por Carmo e Courtiat [CaCo96], de modo a reforçar o poder da sincronização temporal. A técnica de Escobar et al. [EPD94] utiliza um protocolo de sincronização adaptativo a mudanças de atraso de fluxos, capaz de sincronizar aplicações ao vivo. Ramanathan e Rangan [RaRa93] propõem técnicas de realimentação e protocolos para a sincronização intra e intermídia de dados pré-armazenados. Estas técnicas podem ser analisadas através de suas principais características [Rubi96, BlSt96], porém nenhuma delas garante todos os requisitos das várias aplicações existentes.

Neste trabalho é proposta uma extensão ao mecanismo de sincronização de pares de mídias (mestra, escrava) implementado no Grupo de Teleinformática e Automação da UFRJ [Rubi96, RuDu97], de modo a diminuir o *overhead* devido ao envio de uma unidade de realimentação para cada unidade de mídia exibida na escrava. Dada a assincronia máxima permissível entre a mestra e uma escrava de um feixe de sincronização (conjunto de fluxos que possuem relações temporais entre si), um servidor de sincronização determina a menor taxa de transmissão pela escrava de unidades de realimentação, mantendo a sincronização entre os pares de mídias. Esta menor taxa de realimentação diminui substancialmente o processamento no servidor e o *overhead* na rede.

O artigo está organizado da seguinte maneira: a seção 2 apresenta o serviço de sincronização e o seu respectivo protocolo, a seção 3 descreve o mecanismo de sincronização e na seção 4 são apresentadas as conclusões.

2 Camada Sincronização

Para ter-se uma única base de tempo (essencial em sincronização), uma arquitetura centralizada constituída de um Servidor de Sincronização (SS) e de diversas estações multimídias interconectadas por uma rede de computadores é utilizada. Com isto, as diferenças temporais entre os relógios envolvidos na sincronização são facilmente levadas em consideração, através de um mapeamento do relógio de cada fonte/destino no relógio do servidor. A idéia chave da centralização é concentrar a maior carga de processamento no servidor, permitindo a utilização como fontes e destinos de estações multimídias de menor capacidade de processamento e, portanto, de menor custo. Além disto, o servidor permite o atendimento de diversas aplicações ao mesmo tempo, garantindo que dados gerados por fontes distribuídas sejam apresentados de forma sincronizada nos seus destinos.

Em cada feixe de sincronização é definida uma **mídia mestra**, para a qual é garantida a continuidade. Todas as outras mídias de um feixe de sincronização são escravas. As **mídias escravas** são sincronizadas em relação à mestra, estando sujeitas a saltos (para adiantá-las) ou repetições (para atrasá-las), quando a assincronia ultrapassa um valor predeterminado. Os **saltos** correspondem à não exibição de *n* unidades elementares de uma mídia, enquanto que as **repetições** correspondem à reexibição de uma unidade elementar de uma mídia *n* vezes. A unidade elementar de uma mídia será doravante denominada **Unidade de Mídia** (UM). A escolha da mídia mestra depende da aplicação. No caso de uma videoconferência, prioriza-se a mídia áudio em detrimento da mídia vídeo, pois um grande intervalo entre amostras de áudio e/ou uma repetição ou perda de amostras de áudio podem prejudicar a inteligibilidade.

2.1 Serviço de sincronização

O presente trabalho concentra as funções de sincronização em uma camada denominada Sincronização. O serviço está especificado em duas subcamadas: continuidade e sincronização intermídia. Devido à característica de centralização atribuída ao SS, as funções da camada Sincronização são assimétricas. A arquitetura de protocolos utilizada é apresentada na figura 1.

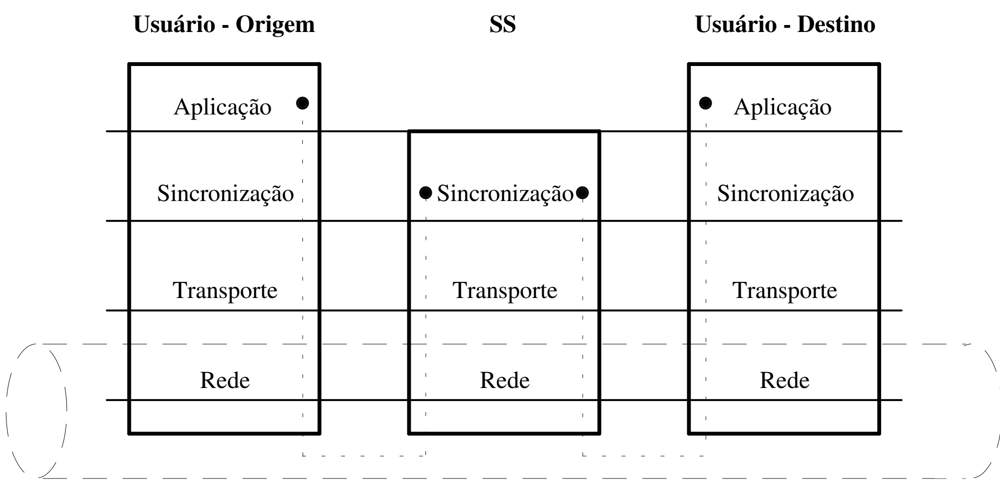

Figura 1: Arquitetura de protocolos utilizada.

Cabe à subcamada continuidade oferecer o serviço de continuidade da mídia mestra, para todos os feixes de sincronização. Esta subcamada utiliza o serviço da camada Transporte, que oferece um serviço de transferência de dados fim-a-fim não confiável e não orientado à conexão.

A subcamada sincronização intermídia utiliza o serviço da subcamada continuidade para prover à aplicação, para cada feixe de sincronização, pares de mídias (*mestra*, *escrava*), cuja assincronia não ultrapassa um valor predeterminado A. Na tabela 1 são apresentadas as primitivas de serviço com uma breve descrição e os parâmetros de cada uma. A primitiva S-JOIN.Request contém os parâmetros de sincronização da seção 3 (delta_min, delta_max, delta_r_min, delta_r_max, theta e ro) e o parâmetro assinc que é a assincronia máxima permissível (A) entre uma escrava e a mestra de um feixe de sincronização. Estes parâmetros são fornecidos pelo usuário, quando na fase de adesão a um feixe. Além disto, o parâmetro opções (seção 2.2) também é utilizado. Todos estes parâmetros estão indicados na tabela 1 como parâmetros de qualidade de serviço.

Tabela 1: Primitivas de serviço de continuidade e sincronização intermídia.

Primitivas	Descrição	Parâmetros
S-CONNECT.*Request*/ S-CONNECT.*Indication*	*Pedido/Indicação de pedido* de estabelecimento de conexão	Endereço destino Endereço fonte Tipo de mídia
S-CONNECT.*Response*/ S-CONNECT.*Confirm*	*Resposta/Confirmação a/de pedido* de estabelecimento de conexão	Endereço respondedor Resposta
S-DISCONNECT.*Request*/ S-DISCONNECT.*Indication*	*Pedido/Indicação de pedido* de liberação de conexão	*Tipo/Razão* de desconexão
S-DATA.*Request*/ S-DATA.*Indication*	*Pedido/Indicação de pedido* de envio de unidade de mídia	Identificador de feixe Tipo de mídia Estampa de Tempo Unidade de realimentação Dados do usuário
S-FEEDBACKED-DATA.*Request*	*Pedido* de envio de unidade de realimentação	Identificador de feixe Estampa de Tempo
S-JOIN.*Request*	*Pedido* de adesão a um feixe	Identificador de fluxo Identificador de feixe Qualidade de serviço
S-JOIN.*Confirm*	*Confirmação de pedido* de adesão a um feixe	Resposta Identificador de fluxo Identificador de feixe
S-LEAVE.*Request*	*Pedido* de desligamento de um fluxo de um feixe	Identificador de fluxo Identificador de feixe
S-ABORT.*Indication*	*Indicação de falta de recursos* para processar um feixe	Código de erro

2.2 Protocolo de sincronização

O protocolo proposto neste trabalho segue a tendência dos protocolos de alta velocidade, onde uma parte do cabeçalho das PDUs (*Protocol Data Units*) e seus campos possuem tamanhos fixos. Todos os campos, do cabeçalho e de dados, são alinhados em quatro *bytes*, de modo a aproveitar o potencial das máquinas de 32 *bits*, bem como para acelerar a decodificação dos campos das PDUs [Nune95]. Assim, todas as PDUs deste protocolo utilizam uma mesma parte fixa do cabeçalho, com três campos de 32 *bits*. Estas características visam minimizar o processamento das PDUs que se constitui na principal limitação de desempenho de um protocolo.

As PDUs são divididas nos seguintes grupos funcionais: conexão, desconexão, transferência de dados, pré-armazenamento, feixe de sincronização e erro. Na tabela 2 são apresentadas as PDUs do protocolo de sincronização separadas nos grupos funcionais com uma breve descrição de suas utilizações.

Tabela 2: Unidades de dados de protocolo de sincronização.

Grupos Funcionais	**PDUs**	**Descrição**
Conexão	**CR** (Connect Request)	Pedido de conexão ponto-a-ponto
	NCR (Numbered Connect Request)	Pedido de conexão numerada ponto-a-ponto
	NCC (Numbered Connect Confirm)	Confirmação de pedido de conexão
Desconexão	**DR** (Disconnect Request)	Pedido de desconexão ponto-a-ponto
Transferência de dados	**DT** (Data)	Envio de dados do usuário
	SD (Stamped Data)	Envio de dados com estampa de tempo relativo
	FD (Feedbacked Data)	Envio de unidade de realimentação
Pré-armazenamento	**PF** (Prefetch)	Pedido de pré-armazenamento de dados
Feixe de sincronização	**JR** (Join Request)	Pedido de adesão de um fluxo a um feixe
	JC (Join Confirm)	Confirmação de adesão de um fluxo a um feixe
	LR (Leave Request)	Pedido de desligamento de um fluxo de um feixe
Erro	**ERROR**	Fornecimento de diagnósticos para diversas situações de erro

As PDUs contém uma parte fixa do cabeçalho e uma parte variável. As PDUs DT e SD são as únicas a possuir segmento de dados. A parte fixa do cabeçalho contém os campos *command*, *key* e *seq*, todos de 32 *bits* (figura 2). O campo *cmd* (*command*), conforme a figura 2, contém informações sobre o tipo da PDU (tipo), o número de *bytes* necessários para completar o campo de dados do usuário, de modo a tornar seu comprimento múltiplo de 32 *bits* (*padding*) e as opções de configuração do protocolo (opções). As opções, CON e AGR, indicam a utilização da política conservativa ou agressiva, respectivamente (seção 3.1.2). Estas opções são utilizadas na PDU de pedido de adesão JR (Join Request). Os *bits* de 18 a 31 estão reservados para futuras extensões. O campo *key* serve para associar uma PDU a uma determinada conexão. Este campo é preenchido pelo SS, que atua como um gerente de conexões de sincronização. O campo *seq* indica o número de seqüência de PDUs DT, SD e FD. Este campo deve conter o valor zero em quaisquer outras PDUs.

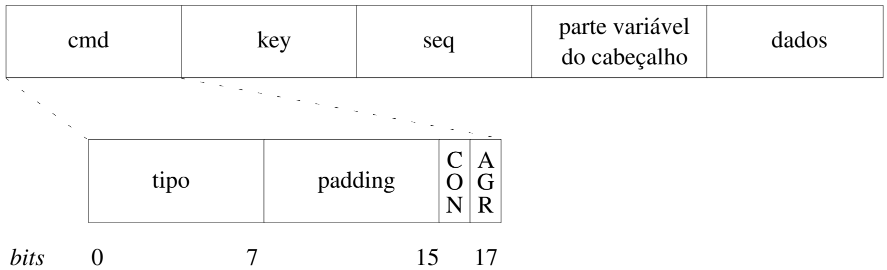

Figura 2: Formato das PDUs e do campo *command*.

3 Mecanismo de Sincronização

O mecanismo de sincronização destina-se a aplicações que possuem as seguintes características:

- envolvem fontes de dados armazenados ou ao vivo, de modo a cobrir todos os tipos de fontes de dados;
- envolvem relações temporais implícitas, pois o objetivo das apresentações é exibir os dados de maneira igual os quais foram capturados;
- envolvem múltiplas fontes e destinos, de modo a cobrir qualquer configuração de estações participantes das aplicações;
- não utilizam relógio global.

As relações temporais implícitas entre os dados são representadas através de **Estampas de Tempo Relativo** (ETRs) que indicam o tempo relativo à geração/exibição da primeira unidade de mídia, de um feixe de sincronização, a chegar ao SS. O SS é responsável por colocar ETRs nas unidades de mídia envolvidas na aplicação e controlar o envio destas unidades de mídia para os seus respectivos destinos, de modo a evitar que os *buffers* nos destinos fiquem vazios (*starvation*) ou cheios (*overrun*).

A técnica de sincronização utilizada [RuDu96, RuDu97, Rubi96], baseia-se na proposta de Rangan et al. [RaRa93, RRK95]. Portanto, assume-se que os relógios não são sincronizados e que o tempo entre a transmissão pela fonte da primeira unidade de dado de uma unidade de mídia e a recepção pelo SS da última unidade de dado desta unidade de mídia está entre $\Delta_{\min}$ e $\Delta_{\max}$. Define-se uma **Unidade de Realimentação** (UR) como sendo uma unidade elementar utilizada para controlar o envio de unidades de mídia para os seus respectivos destinos. Assim, o tempo entre a transmissão pelo destino da primeira unidade de dado de uma unidade de realimentação e a recepção pelo SS da última unidade de dado desta unidade de realimentação está entre $\Delta^r_{\min}$ e $\Delta^r_{\max}$. Apesar da técnica não utilizar relógios sincronizados, as variações nas taxas de crescimento dos tempos dos relógios (*clock drift*) de diferentes estações são levadas em conta, através das estimativas do menor e maior tempos de duração de uma unidade de mídia. Sendo θ a duração nominal de uma unidade de mídia e ρ a variação máxima fracional em relação a θ, a duração real de uma unidade de mídia *n*, $\theta(n)$, estará entre $\theta \times (1-\rho)$ e $\theta \times (1+\rho)$, pois $\rho = |\theta(n) - \theta| / \theta$.

Dados gerados por diversas fontes em um determinado instante terão a garantia de uma apresentação simultânea em diferentes destinos. De uma forma simplificada, tem-se as fontes gerando UMs que devem ser encaminhadas aos seus respectivos destinos. O SS coloca estampas de tempo relativo nas UMs baseado no tempo de transferência estimado entre a fonte e o SS, e escalona estas UMs para transmissão para os destinos. Os destinos determinam os tempos relativos e exibem as informações de uma forma sincronizada.

Para cada feixe de sincronização, as principais funções executadas pelo mecanismo de sincronização são: a colocação de estampas de tempo relativo, a garantia da continuidade da mídia mestra e a exibição síncrona. As funções colocação de estampas de tempo e exibição síncrona são calculadas para todas as escravas de um feixe de sincronização.

Na colocação de estampas de tempo relativo, o SS se encarrega de "gravar" ETRs em todas as UMs e o(s) destino(s) deve(m) apresentar simultaneamente as UMs que possuem um mesmo valor de ETR. A ETR de geração de uma unidade de mídia mestra n_m é o seu número de seqüência n_m, enquanto que a ETR de geração de uma unidade de mídia escrava n_e é o número de seqüência da unidade de mídia da mestra que poderia ter sido gerada ao mesmo tempo que n_e. Devido à presença de atrasos não determinísticos da rede e de diferentes taxas de captura das fontes, o SS não pode determinar precisamente os tempos de geração de UMs e, portanto, só pode ser capaz de assinalar intervalos de ETRs para UMs da escrava. Sempre que o SS recebe uma unidade de mídia n_e, este determina o intervalo de geração de n_e (relativo à mestra) utilizando a última UM da mestra recebida antes de n_e (denominada n_m). Com este intervalo, junto ao intervalo de geração de n_m, o SS calcula quais seriam o menor e o maior números de seqüência cujas UMs correspondentes da mestra poderiam ter sido geradas concorrentemente com n_e. O valor que melhor representa a ETR de n_e é calculado e utilizado no destino para a exibição de n_e [Rubi96].

Para garantir a continuidade da mídia mestra, de acordo com [RaRa93], calcula-se para cada UM *n* da mestra o intervalo entre as transmissões da P-ésima UM (supondo um pré-armazenamento *P*) e da UM *n*, ou seja, um *deadline* para a transmissão da UM *n*. Para isto, utiliza-se um pré-armazenamento de UMs e unidades de realimentação que são transmitidas pela mestra quando da exibição de certas UMs. Uma UR contém a ETR da UM correspondente. Um *flag* binário, habilitado pelo SS antes do começo da exibição, é utilizado para a mestra saber se deve transmitir uma UR para a UM exibida. O cálculo dos tempos mínimo e máximo de transmissão das UMs a serem enviadas após a recepção de uma UR e o cálculo da menor taxa de envio de URs, garantindo-se a continuidade da mídia mestra, são apresentados detalhadamente em [Rubi96].

Na função de exibição síncrona apresentada em [Rubi96], para cada unidade de mídia da escrava apresentada é enviada uma unidade de realimentação ao SS, para que o SS verifique como está a exibição naquele destino. A seguir é apresentado o cálculo da menor taxa de envio de URs pela escrava, uma modificação na função de exibição síncrona, de modo a diminuir o *overhead* devido à realimentação.

3.1 Exibição Síncrona

Para que a sincronização intermídia seja mantida é feito um cálculo do intervalo das ETRs de exibição de certas UMs da escrava, utilizando URs transmitidas concorrentemente com o início da exibição destas UMs, diferentemente de [RRK95]. Com este intervalo e o intervalo das ETRs de geração da mesma UM, é verificada se a assincronia da escrava em relação à mestra ultrapassa um valor predeterminado, sendo tomadas ações para sincronizar as mídias (seção 3.1.2). A ETR de exibição de uma unidade de mídia n_e da escrava é o número de seqüência da unidade de mídia da mestra que poderia ter sido exibida ao mesmo tempo que n_e.

Devido à presença dos atrasos de rede não determinísticos e das diferentes taxas de exibição dos destinos, o SS também não é capaz de determinar precisamente os tempos de exibição de UMs nos destinos, e portanto, só pode assinalar intervalos de ETRs para UMs da escrava. Sempre que o SS recebe uma unidade de realimentação da escrava (transmitida concorrentemente com a exibição da unidade de mídia n_e), este determina o intervalo de exibição de n_e. Com este intervalo, junto ao intervalo de exibição de n_m, o SS calcula quais seriam o menor e o maior números de seqüência cujas UMs correspondentes da mestra poderiam ter sido exibidas concorrentemente com n_e. Com isto, o SS obtém o intervalo das ETRs de exibição de n_e. Todos os cálculos seguintes utilizam o superescrito - para indicar o menor tempo de um evento ("o mais cedo que o evento pode ocorrer") e o superescrito + para indicar o maior tempo deste evento ("o mais tarde que o evento pode ocorrer"). Como exemplo, $t^-(n)$ indica o menor tempo de exibição de uma unidade de mídia n.

Sendo $t_e(n)$ o tempo da exibição da unidade de mídia n e $\tau_e(n)$ o tempo da chegada ao SS da unidade de realimentação transmitida concorrentemente com a exibição da unidade de mídia n, o SS determina o menor e o maior tempos de exibição das unidades de mídia n_m e n_e, utilizando tempos de chegada ao SS das unidades de realimentação relativas a n_m e n_e e os atrasos de realimentação $\Delta^r_{\min}$ e $\Delta^r_{\max}$, da seguinte maneira:

$$t_e^-(n_m) = \tau_e(n_m) - \Delta^r_{\max}\,, \tag{1}$$

$$t_e^+(n_m) = \tau_e(n_m) - \Delta^r_{\min}\,, \tag{2}$$

$$t_e^-(n_e) = \tau_e(n_e) - \Delta^r_{\max} \text{ e} \tag{3}$$

$$t_e^+(n_e) = \tau_e(n_e) - \Delta^r_{\min}\,. \tag{4}$$

Os intervalos $[t_e^-(n_m), t_e^+(n_m)]$ e $[t_e^-(n_e), t_e^+(n_e)]$ podem estar superpostos ou não. O caso quando não há superposição dos intervalos (figura 3) é tratado a seguir. Os cálculos para o caso de haver superposição dos intervalos são feitos de maneira análoga.

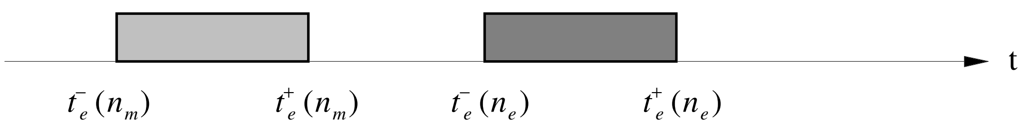

Figura 3: Diagrama temporal dos tempos de exibição de n_m e n_e.

Sendo $n_m+\upsilon^-$ o menor número de seqüência cuja UM da mestra poderia ter sido exibida concorrentemente com n_e e $n_m+\upsilon^+$ o maior número de seqüência cuja UM da mestra poderia ter sido exibida concorrentemente com n_e, para a UM $n_m+\upsilon^-$ ter sido exibida concorrentemente com n_e, de acordo com a figura 4, a seguinte inequação deve ser satisfeita:

$$t_e^+(n_m + \upsilon^-) \geq t_e^-(n_e) \tag{5}$$

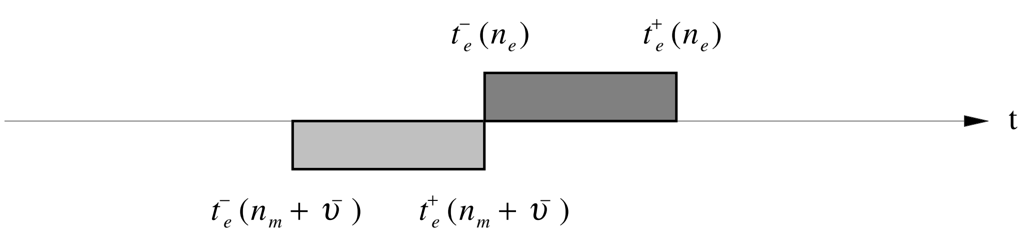

Figura 4: Diagrama temporal para o cálculo de υ^-.

Já para a UM $n_m+\upsilon^+$ ter sido exibida concorrentemente com n_e, de acordo com a figura 5, a seguinte inequação deve ser satisfeita:

$$t_e^-(n_m + \upsilon^+) \leq t_e^+(n_e)\,. \tag{6}$$

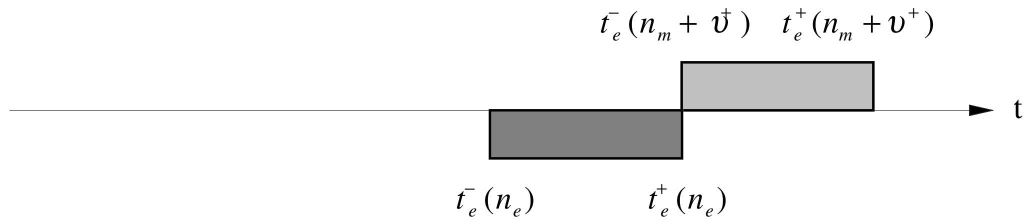

Figura 5: Diagrama temporal para o cálculo de υ^+.

Como a mídia mestra deve ser contínua, sabe-se que:

$$t_e^-(n_m+\upsilon^+) = t_e^-(n_m) + \upsilon^+ \times \theta \times (1-\rho) \text{ e} \tag{7}$$

$$t_e^+(n_m+\upsilon^-) = t_e^+(n_m) + \upsilon^- \times \theta \times (1+\rho)\,, \tag{8}$$

e utilizando-se (5), (6), (7) e (8), tem-se:

$$t_e^+(n_e) \geq t_e^-(n_m) + \upsilon^+ \times \theta \times (1-\rho) \text{ e}$$

$$t_e^-(n_e) \leq t_e^+(n_m) + \upsilon^- \times \theta \times (1+\rho)\,.$$

Como υ^- e υ^+ devem ser inteiros, tem-se[2]:

$$\upsilon^- = \left\lceil \frac{t_e^-(n_e) - t_e^+(n_m)}{\theta \times (1+\rho)} \right\rceil \text{ e} \tag{9}$$

$$\upsilon^+ = \left\lfloor \frac{t_e^+(n_e) - t_e^-(n_m)}{\theta \times (1-\rho)} \right\rfloor. \tag{10}$$

O intervalo das ETRs de n_e é dado por $[e_e^-(n_e), e_e^+(n_e)]$, onde $e_e^-(n_e) = n_m + \upsilon^-$ e $e_e^+(n_e) = n_m + \upsilon^+$.

3.1.1 Cálculo da menor taxa de envio de unidades de realimentação

Para garantir a exibição síncrona de pares de unidades de mídia da mestra e da escrava, são enviadas URs pela escrava quando da exibição de certas UMs. Um flag binário, setado pelo SS antes do começo da exibição, é utilizado para saber se a escrava deve transmitir uma UR para a UM exibida. Não há necessidade de enviar URs enquanto a assincronia não ultrapassar *A* UMs. O cálculo da menor taxa de envio de URs de uma escrava supõe que para cada par (n_e, n_m), a assincronia máxima pode ser estabelecida. Neste caso uma das UMs sofrerá um atraso do menor valor e será exibida no menor tempo, enquanto que a outra sofrerá um atraso do maior valor e será exibida no maior tempo. Com isto, a assincronia máxima é:

$$a = (\Delta_{\max} + \theta \times (1+\rho)) - (\Delta_{\min} + \theta \times (1-\rho))\,, \tag{11}$$

$$a = (\Delta_{\max} - \Delta_{\min}) + 2 \times \theta \times \rho \text{, onde}$$

$$\Delta_{\max} - \Delta_{\min} = \max(|\Delta_{\max}^m - \Delta_{\min}^e|, |\Delta_{\max}^e - \Delta_{\min}^m|)\,.$$

Para *n* UMs, a assincronia máxima deverá ser menor ou igual a *A* UMs, logo:

$$((\Delta_{\max} - \Delta_{\min}) + 2 \times \theta \times \rho) \times n \leq A \times \theta \times (1-\rho)\,, \tag{12}$$

$$n \leq \frac{A \times \theta \times (1-\rho)}{((\Delta_{\max} - \Delta_{\min}) + 2 \times \theta \times \rho)}\,. \tag{13}$$

Sendo, $F = \dfrac{1}{n}$, a taxa de envio de unidades de realimentação pela escrava, tem-se:

---

[2]Os símbolos $\lfloor a \rfloor$ e $\lceil a \rceil$ significam, respectivamente, o maior inteiro $\leq a$ e o menor inteiro $\geq a$.

$$F = \frac{((\Delta_{\max} - \Delta_{\min}) + 2 \times \theta \times \rho)}{A \times \theta \times (1-\rho)}. \tag{14}$$

Com isto, deve ser enviada uma unidade de realimentação a cada

$$\left\lfloor \frac{1}{F} \right\rfloor = \left\lfloor \frac{A \times \theta \times (1-\rho)}{((\Delta_{\max} - \Delta_{\min}) + 2 \times \theta \times \rho)} \right\rfloor \tag{15}$$

unidades de mídia da escrava.

Os valores utilizados no protótipo de sistema de sincronização que encontra-se em funcionamento no Laboratório do GTA podem ser empregados de modo a avaliar o desempenho da extensão proposta. O protótipo roda em estações SUN-SPARC 20, com sistema operacional Solaris 2.4, ligadas por uma rede Ethernet. Supondo um vídeo JPEG a 10 quadros/s como mídia escrava, tem-se $\theta = 10ms$, $\Delta^{e}_{\min} = 6ms$ e $\Delta^{e}_{\max} = 12ms$. Um áudio a 64kbps é utilizado como mídia mestra e os atrasos são $\Delta^{m}_{\min} = 2ms$ e $\Delta^{m}_{\max} = 3ms$. Sendo a assincronia máxima permissível entre o áudio e o vídeo de 80ms conforme [Stei96], a assincronia máxima em número de UMs deve ser $A = 8$. A variação máxima fracional em relação a θ é $\rho = 10^{-4}$, ou seja, correponde à variação de algumas dezenas de segundos por dia. Com estes valores, conforme (15), tem-se uma UR a cada 7 UMs, ou seja, uma taxa de envio de unidades de realimentação sete vezes menor, diminuindo-se substancialmente o processamento no SS e o *overhead* na rede.

3.1.2 Políticas de sincronização

A assincronia pode existir mesmo se os intervalos das ETRs de geração e de exibição forem iguais. Garante-se que a assincronia existe somente quando os dois intervalos não se superpõem. A ressincronização pode ser forçada se há certeza ou mesmo uma pequena chance detectável de assincronia, surgindo as políticas de ressincronização conservativa e agressiva.

Sejam $[e_g^-(n_e), e_g^+(n_e)]$ e $[e_e^-(n_e), e_e^+(n_e)]$ os intervalos de estampas de tempo relativo de geração e exibição, respectivamente, da UM n_e. Sendo a máxima assincronia possível da escrava em relação à mestra, no tempo de exibição de n_e, $\delta e_{\max}(n_e) = \max(|e_g^+(n_e) - e_e^-(n_e)|, |e_g^-(n_e) - e_e^+(n_e)|)$, tem-se:

- a assincronia da escrava não excede *A*, se e somente se $|e_g^+(n_e) - e_e^-(n_e)| \le A$ **e** $|e_g^-(n_e) - e_e^+(n_e)| \le A$, ou seja, se $\delta e_{\max}(n_e) \le A$;
- a assincronia da escrava excede *A*, se e somente se $e_g^-(n_e) - e_e^+(n_e) > A$ **ou** $e_e^-(n_e) - e_g^+(n_e) > A$.

A política conservativa atua quando a assincronia excede A. Se a escrava está atrasada em relação à mestra ($e_e^-(n_e) - e_g^+(n_e) > A$), o número de UMs descartadas é $e_e^-(n_e) - e_g^+(n_e)$ (figura 6). Quando a escrava está adiantada em relação à mestra ($e_g^-(n_e) - e_e^+(n_e) > A$), uma UM será repetida $e_g^-(n_e) - e_e^+(n_e)$ vezes.

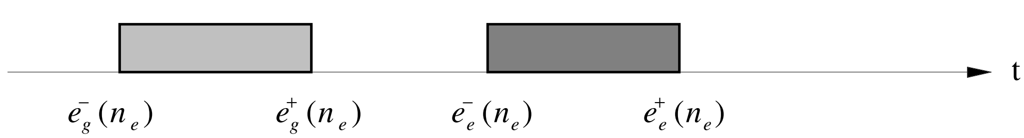

Figura 6: Diagrama temporal mostrando a escrava atrasado em relação à mestra.

A política agressiva atua quando há uma chance da assincronia exceder A, ou seja, $|e_g^+(n_e)-e_e^-(n_e)|>A$ **ou** $|e_g^-(n_e)-e_e^+(n_e)|>A$, logo se $\delta e_{\max}(n_e)>A$. Se $|e_g^+(n_e)-e_e^-(n_e)|>|e_g^-(n_e)-e_e^+(n_e)|$, $|e_g^-(n_e)-e_e^+(n_e)|$ UMs serão repetidas, caso contrário, $|e_g^+(n_e)-e_e^-(n_e)|$ UMs serão descartadas.

De acordo com Rangan et al. [RRK95], a política agressiva exibe grande reatividade e comportamento oscilatório quando a assincronia A é pequena, mas melhora muito seu desempenho quando a assincronia A é grande. A política conservativa atua com uma maior eficiência quando a assincronia A é pequena e não atua com grande eficiência quando a assincronia A é grande. Isto pôde ser comprovado no funcionamento do protótipo do sistema de sincronização, onde não ocorrem grandes níveis de assincronia, ou seja, a política conservativa praticamente não atua e a política agressiva exibe uma maior reatividade e uma grande assincronia média [Teix97].

4 Conclusões

Este artigo apresentou um serviço de sincronização definido através de suas primitivas e o seu respectivo protocolo. Este sistema foi implementado utilizando-se um ambiente de desenvolvimento de sistemas de comunicação de alto desempenho com garantias de qualidade de serviço denominado Arquitetura GTA/MM.

O serviço de sincronização oferecido baseia-se em um mecanismo apropriado para aplicações multimídias que envolvam fontes de dados armazenados ou ao vivo, relações temporais implícitas, múltiplas fontes e destinos e não utilizem relógio global. Este mecanismo utiliza uma arquitetura centralizada constituída de um servidor de sincronização e de diversas estações multimídias interconectadas através de uma rede de computadores.

Neste trabalho foi proposta uma extensão ao mecanismo de sincronização de pares de mídias (*mestra*, *escrava*), de modo a diminuir o *overhead* devido ao envio de uma unidade de realimentação para cada unidade de mídia exibida na escrava. Esta nova técnica baseia-se no fato de não haver necessidade de envio de unidades de realimentação enquanto a assincronia não ultrapassar um valor predeterminado. O cálculo da menor taxa de envio de unidades de realimentação de uma escrava supõe que a assincronia máxima pode ser estabelecida, considerando que uma das unidades de mídia sofre um atraso do menor valor e é exibida no menor tempo, enquanto que a outra sofre um atraso do maior valor e é exibida no maior tempo. Para valores utilizados no protótipo do sistema de sincronização do GTA, esta técnica possibilita uma taxa de envio de uma unidade de realimentação para cada sete unidades de mídia exibidas na escrava, ou seja, uma taxa de realimentação sete vezes menor, diminuindo substancialmente o processamento no servidor e o *overhead* na rede.

Referências

[BlSt96] G. Blakowski e R. Steinmetz, "A Media Synchronization Survey: Reference Model, Specification, and Case Studies", *IEEE Journal on Selected Areas in Communications*, vol. 14, nº 1, pp. 5-35, janeiro de 1996.

[CaCo96] L. F. R. C. Carmo e J.-P. Courtiat, "Extrator de Dependências Condicionais para Sistemas de Comunicação Multimídias", *XIV Simpósio Brasileiro de Redes de Computadores*, Fortaleza, pp. 56-73, maio de 1996.

[EPD94] J. Escobar, C. Partridge e D. Deutsch, "Flow Synchronization Protocol", *IEEE/ACM Transactions on Networking*, vol. 2, nº 2, pp. 111-121, abril de 1994.

[LiDu96] R. M. de Lima e O. C. M. B. Duarte, "Comunicação Sincronizada de Dados Multimídias Pré-Orquestrados", *XXIII Seminário Integrado de Software e Hardware* (*SEMISH'96*), Recife, pp. 285-296, agosto de 1996.

[LiGh90] T. D. C. Little e A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects", *IEEE Journal on Selected Areas in Communications*, vol. 8, nº 3, pp. 413-426, abril de 1990.

[LiGh91] T. D. C. Little e A. Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services", *IEEE Journal on Selected Areas in Communications*, vol. 9, nº 9, pp. 1368-1382, dezembro de 1991.

[Nune95] M. D. Nunes, "Protocolo Multidestinatário com Configuração de Facilidades e Gerenciamento de Acesso", *Tese de Mestrado*, PEE-COPPE/UFRJ, abril de 1995.

[RaRa93] S. Ramanathan e P. V. Rangan, "Feedback Techniques for Intra-Media Continuity and Inter-Media Synchronization in Distributed Multimedia Systems", *The Computer Journal*, vol. 36, nº 1, pp. 19-31, fevereiro de 1993.

[RRK95] P. V. Rangan, S. Ramanathan e T. Kaeppner, "Performance of Inter-Media Synchronization in Distributed and Heterogeneous Multimedia Systems", *Computer Networks and IDSN Systems*, vol. 27, pp. 549-565, 1995.

[Rubi96] M. G. Rubinstein, "Um Serviço de Sincronização Contínua para Aplicações Multimídias", *Tese de Mestrado*, PEE-COPPE/UFRJ, novembro de 1996.

[RuDu96] M. G. Rubinstein e O. C. M. B. Duarte, "Definição de um Serviço de Sincronização Contínua para Aplicações Multimídias", *XIV Simpósio Brasileiro de Redes de Computadores*, Fortaleza, pp. 38-55, maio de 1996.

[RuDu97] M. G. Rubinstein e O. C. M. B. Duarte, "Implementação de um Serviço de Sincronização Baseado em Relações Temporais Implícitas", *II Seminário Franco-Brasileiro em Sistemas Informáticos Distribuídos*, Fortaleza, pp. 282-293, novembro de 1997.

[Stei96] R. Steinmetz, "Human Perception of Jitter and Media Synchronization", *IEEE Journal on Selected Areas in Communications*, vol. 14, nº 1, pp. 61-72, janeiro de 1996.

[Teix97] R. C. Teixeira, "Um Serviço de Sincronização Interfluxo para Comunicação Multimídia", *Projeto de Fim de Curso*, DCC-IM/UFRJ, fevereiro de 1997.