Definição de um Serviço de Sincronização Contínua para Aplicações Multimídias[1]

Marcelo Gonçalves Rubinstein[2]
rubi@coe.ufrj.br

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

Grupo de Teleinformática e Automação - GTA
Universidade Federal do Rio de Janeiro
COPPE/EE - Programa de Engenharia Elétrica
Caixa Postal 68504 - CEP 21945-970 - Rio de Janeiro - RJ - Brasil
FAX: +55 21 2906626

Resumo

A transmissão de várias mídias em redes de computadores requer a utilização de mecanismos especiais que possibilitem apresentações multimídias coerentes. A sincronização em aplicações multimídias é necessária quando a cadência de um fluxo isócrono e/ou a relação temporal entre fluxos isócronos ou não são perdidas. Este trabalho descreve um serviço de continuidade e sincronização intermídia, bem como o respectivo protocolo, apropriado para aplicações multimídias que envolvam fontes de dados armazenados e/ou ao vivo, relações temporais implícitas, múltiplas fontes e destinos e não utilizem relógio global. O serviço é definido através das suas primitivas de serviço. O tipo e o formato das unidades de dado de protocolo e as características da implementação relativas à arquitetura e ao ambiente de desenvolvimento são apresentados.

Abstract

The transmission of several media on computer networks requires the use of special mechanisms, well suited for coherent multimedia presentations. Synchronization must be employed when isochronous data flow continuity and/or temporal relationship between isochronous or non isochronous data flows are lost. This paper describes a continuity and intermedia synchronization service, and respective protocol, well suited for multimedia applications that involves stored and/or live data sources, implicit temporal relationships, multiple sources and destinations and do not utilize global clock. Service is defined by its service primitives. Protocol data units' type and format and implementation characteristics related to architecture and development environment are presented.

[1] Este trabalho foi realizado com recursos da UFRJ, FUJB, CNPq, PROTEM-CC e CAPES.
[2] Aluno de Mestrado do Programa de Engenharia Elétrica da COPPE/UFRJ.

1 Introdução

A partir do aumento da banda passante das redes de telecomunicações, dos avanços na tecnologia de armazenamento de mídias e do aprimoramento dos equipamentos de áudio e vídeo digitais, novas aplicações envolvendo a integração de várias mídias apareceram, surgindo a Multimídia. Novos requisitos de Qualidade de Serviço (QoS) foram introduzidos, tais como o sincronismo entre as amostras de áudio e entre os quadros de vídeo.

Nos últimos anos, várias técnicas de sincronização foram propostas. O servidor de Anderson & Homsy [AnHo91] utiliza algoritmos para sincronizar aparelhos de entrada e saída de mídias contínuas em uma única estação rodando um sistema operacional de tempo real. Shepherd et al. [SHGC92] apresentam mecanismos de transporte de informações multimídias em um sistema distribuído. Uma arquitetura para sistemas multimídias em tempo real na qual pontos de sincronização são utilizados para sincronizar fluxos relacionados é vista em Nicolaou [Nico90]. No esquema de Ferrari [Ferr92], é descrito um mecanismo distribuído para controlar a variação do atraso (*jitter*) em redes que têm os comutadores e as operações de comutação muito complicadas. O modelo de Little & Ghafoor [LiGh91] utiliza Redes de Petri para a Composição de Objetos (OCPNs) para especificar as relações temporais entre os objetos e um algoritmo para a recuperação de dados e identificação de tamanhos de *buffers* necessários. Um mecanismo de sincronização intermídia que utiliza dependências condicionais para expressar pontos de sincronização é proposto por Courtiat & Carmo [CoCa93]. A técnica de Escobar et al. [EPD94] utiliza um protocolo de sincronização adaptativo a mudanças de atraso de fluxos, baseado na utilização de relógios sincronizados e capaz de sincronizar aplicações ao vivo. Ramanathan & Rangan [RaRa93] propõem técnicas de realimentação e protocolos para as sincronizações intra e intermídia de dados pré-armazenados, na presença de variação de atraso da rede e de diferenças não determinísticas nas taxas de captura e/ou exibição de mídias de diferentes estações. Nenhuma das técnicas garante todos os requisitos das várias aplicações existentes [EFI94]. Um estudo da aplicação a ser desenvolvida deve ser feito, junto com as disponibilidades de *hardware*, a fim de se escolher a técnica a utilizar.

Este trabalho descreve um serviço de continuidade e sincronização intermídia, bem como o respectivo protocolo, apropriado a aplicações multimídias que envolvam fontes de dados armazenados e/ou ao vivo, relações temporais implícitas, múltiplas fontes e destinos e não utilizem relógio global. O serviço provê a sincronização intramídia para as mídias mestras de todos os feixes de sincronização e a sincronização intermídia, para todos os feixes de sincronização, entre os pares (mestra, escrava) para todas as escravas de um feixe de sincronização. As características da implementação relativas à arquitetura e ao ambiente de desenvolvimento são apresentadas.

O trabalho está organizado da seguinte maneira: a seção 2 apresenta os principais conceitos básicos relacionados à sincronização, a seção 3 descreve o serviço de continuidade e sincronização intermídia proposto, a seção 4 apresenta as primitivas do serviço de continuidade e sincronização intermídia, a seção 5 descreve o tipo e o formato das unidades de dados de protocolo de continuidade e sincronização intermídia e a seção 6 apresenta as características da arquitetura e do ambiente de desenvolvimento da implementação.

2 Conceitos Básicos

Multimídia pode ser definida [Stei90] como a geração, a representação, o processamento, o armazenamento e a disseminação integrados de informações expressas por meio de múltiplas mídias dependentes e independentes do tempo. A unidade elementar de uma mídia (dado) é definida como unidade de mídia. A unidade de mídia pode ser uma amostra de som, um quadro de vídeo, uma página de texto e outros. Não existe, necessariamente, uma relação entre a unidade de mídia e a unidade de transmissão. Uma unidade de dados (exemplos: pacote, célula ou quadro) pode conter várias unidades de mídia assim como podem ser necessárias várias unidades de dados para transferir uma unidade de mídia. Assim, o tempo de transferência (ou atraso) relativo a uma unidade de mídia deve ser computado levando em conta o somatório dos tempos de transferência das várias unidades de dados que constituem uma unidade de mídia. Define-se fluxo de mídia, ou simplesmente fluxo, como a seqüência de unidades de dados referentes a uma mídia transferida pela rede. Define-se fluxo isócrono como o fluxo que requer uma apresentação contínua de suas unidades de dados a intervalos fixos de tempo. Um conjunto de fluxos que possuem relações temporais entre si é denominado "feixe de sincronização". Define-se unidade de realimentação como uma unidade elementar utilizada para controlar o envio de unidades de mídia para os seus respectivos destinos, de modo a evitar que *buffers* nos destinos fiquem vazios (*starvation*) ou cheios (*overrun*).

Os dispositivos dos quais são obtidos os dados de uma aplicação multimídia são classificados em dois tipos: fontes de dados ao vivo e fontes de dados armazenados. As fontes de dados ao vivo são dispositivos que capturam os eventos do meio externo, convertendo-os em dados para o sistema. Estas fontes geram dados em tempo real, na mesma taxa em que são requeridos para consumo. Exemplos deste tipo de fonte são o microfone e a câmera de vídeo. As fontes de dados armazenados são dispositivos que armazenam os dados capturados, podendo gerar dados a taxas diferentes das quais foram capturados e/ou serão apresentados. Exemplos deste tipo de fonte são o disco rígido e o CD-ROM.

A utilização da sincronização em aplicações multimídias é necessária quando a cadência de um fluxo isócrono e/ou a relação temporal entre fluxos isócronos ou não são perdidas. A sincronização pode ser classificada em relação ao fluxo em: sincronização intramídia (intrafluxo) e sincronização intermídia (interfluxo). A sincronização intramídia, também chamada continuidade, corresponde à relação temporal existente entre um dado e o próximo dado do mesmo fluxo, sendo este fluxo isócrono ou não. O intervalo entre a apresentação de dois dados seqüenciais de um mesmo fluxo isócrono 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.

Uma outra classificação para a sincronização é feita segundo a forma de se expressar as relações temporais. A sincronização pode ser: contínua [SHGC92] ou sintética [Esco95]. A sincronização contínua, também chamada sincronização natural [SoAg94] ou sincronização ao vivo [Esco95], mantém relações temporais implícitas de modo contínuo, para uma ou mais mídias. Estas relações são expressas através da relação temporal *igual*. Como exemplo, tem-se a sincronização labial entre a voz e o vídeo. A sincronização sintética, também chamada sincronização baseada em eventos [SHGC92], cria relações temporais explícitas entre objetos

multimídias. As relações temporais podem ser do tipo *antes*, *depois*, *durante* e outras, num total de treze [LiGh90]. Como exemplo, a relação A *antes* T, com um parâmetro X, pode indicar que a apresentação de um *slide* A deve ocorrer X ms antes do começo da apresentação do texto explicativo T. As aplicações que se servem dos mecanismos de sincronização sintética são chamadas de pré-orquestradas [LiDu96].

Outro tipo de sincronização existente é a espacial. A noção de composição espacial corresponde à combinação de objetos para a apresentação em um dispositivo de saída, em um dado instante de tempo. A composição espacial pode envolver certas operações como mudanças de escala, corte, conversão de cores e disposição dentro de uma janela [SCTN92]. A sincronização espacial não é objeto de estudo deste artigo.

Neste trabalho será usada uma abstração denominada "aparelho de mídia" que se refere de uma maneira geral à captura, exibição, processamento, transmissão e recepção de uma mídia. Assim, um computador pessoal que captura áudio através de um microfone e de uma placa digitalizadora e transmite esta informação é um aparelho de áudio. Uma estação de trabalho que recebe informações de vídeo, descompacta-as e as exibe é um aparelho de vídeo. Os aparelhos de mídia que transmitem informações através da rede são denominados fontes, enquanto que os aparelhos de mídia que recebem informações da rede são denominados destinos. Desta forma, em uma mesma estação podem existir mais de uma fonte e/ou destino.

3 Serviço de Sincronização Proposto

O serviço de sincronização proposto neste trabalho destina-se a aplicações que geram dados ao vivo ou armazenados e que possuem relações temporais implícitas. Em outras palavras, dados gerados por diversas fontes em um determinado instante terão a garantia de uma apresentação simultânea em diferentes destinos. Este serviço baseia-se em uma arquitetura centralizada constituída de um Servidor de Sincronização (SS) e diversos aparelhos de mídia interconectados por uma rede de computadores. A idéia chave da centralização é concentrar a maior carga de processamento no Servidor de Sincronização permitindo aparelhos de mídia mais simples e, portanto, de menor custo. Além disto, o Servidor de Sincronização permite atender diversas aplicações ao mesmo tempo, garantindo que dados gerados por fontes distribuídas sejam apresentados de uma forma sincronizada nos seus destinos. As mídias a serem apresentadas nos destinos são transferidas em forma de fluxos independentes, onde a relação temporal entre as mídias é obtida pela criação de feixes de sincronização contendo uma mídia mestra como referência para a sincronização da apresentação.

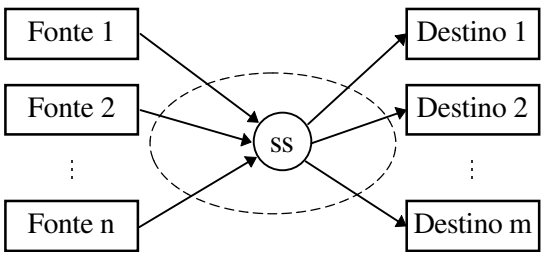

Figura 1: A arquitetura do sistema.

As relações temporais implícitas 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 Servidor de Sincronização é responsável por colocar ETRs nas unidades de mídia envolvidas na aplicação, controlar o envio destas unidades de mídia para os seus respectivos destinos, de modo a evitar que *buffers* nos destinos fiquem vazios (*starvation*) ou cheios (*overrun*), e atualizar uma tabela de fluxos a fim de permitir a existência de vários feixes de sincronização.

A técnica de sincronização utilizada 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 Δ_{min} e Δ_{max}. O tempo entre a transmissão pelo destino da primeira unidade de dado de uma unidade de realimentação e a última unidade de dado desta unidade de realimentação está entre Δ^r_{min} e Δ^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 da aplicação. 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, $\theta(n)$, estará entre $\theta \times (1-\rho)$ e $\theta \times (1+\rho)$, pois $\rho = |\theta(n) - \theta| / \theta$.

Em cada feixe de sincronização é definida uma mídia mestra, para a qual é garantida a continuidade. A mídia mestra de um feixe de sincronização pode ser mudada durante a sincronização. As mídias escravas sincronizam-se em relação à mestra, estando sujeitas a saltos (para apressá-las) ou repetições (para atrasá-las), quando a assincronia ultrapassa um valor predeterminado. Os saltos correspondem à não exibição de n unidades de mídia, enquanto que as repetições correspondem a reexibição de uma mesma unidade de mídia n vezes. 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.

Há a possibilidade de adesão/desligamento de um fluxo de um feixe de sincronização durante a apresentação síncrona. No caso da aplicação envolver somente fontes de dados armazenados, algumas funções oferecidas por aparelhos de videocassete também são fornecidas pelo serviço de sincronização. Há a possibilidade de pausar, retomar, avançar e retroceder a exibição. Isto não é possível para as fontes de dados ao vivo, devido a não existência de um pré-armazenamento das unidades de mídia de uma aplicação. Estas fontes não têm a capacidade de guardar grandes quantidades de unidades de mídia, diferentemente do SS para o caso de fontes de dados armazenados que grava todas as unidades de mídia e as suas respectivas estampas de tempo relativo em discos rígidos.

De uma forma simplificada, tem-se as fontes gerando unidades de mídia que devem ser encaminhadas aos seus respectivos destinos. O SS coloca estampas de tempo relativo nas unidades de mídia baseado no tempo de transferência estimado entre a fonte e o SS, organiza os dados que constituem um feixe de sincronização com relações temporais e garante a

continuidade para a mídia mestra de um feixe de sincronização. O destino determina o tempo relativo e exibe as informações de uma forma sincronizada.

Para cada feixe de sincronização, as principais funções executadas pelo mecanismo de sincronização são: a colocação de estampas, a garantia da continuidade da mídia mestra e a exibição síncrona.

3.1 Colocação das Estampas de Tempo Relativo

O SS se encarrega de "gravar" ETRs em todas as unidades de mídia (UMs) e o destino deve apresentar simultaneamente as unidades de mídia com o mesmo valor de ETR.

A seguir tem-se as definições de ETRs de geração para unidades de mídia do mestre e do escravo, respectivamente[3] :

Definição 1: A ETR de geração de uma unidade de mídia n_m é o seu número de seqüência n_m.

Definição 2: A ETR de geração de uma unidade de mídia n_e é o número de seqüência da unidade de mídia do mestre que poderia ter sido gerada ao mesmo tempo que n_e.

Devido à presença de atrasos de rede não determinísticos e diferentes taxas de captura e/ou exibição de aparelhos de mídia, o SS não deve ser capaz de determinar precisamente os tempos de geração de unidades de mídia nos aparelhos de mídia, e portanto, só pode ser capaz de assinalar intervalos de ETRs (ao invés de um único valor de ETR) para unidades de mídia do escravo [RRK95].

Sempre que o SS recebe uma unidade de mídia n_e, este determina o intervalo de geração de n_e (relativo ao mestre) utilizando a última unidade de mídia do mestre recebida antes de n_e - 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 do mestre poderiam ter sido geradas concorrentemente com n_e. Com isto, o SS obtém o intervalo das ETRs de n_e. É feito ainda o cálculo do valor que melhor representa a ETR de n_e - cálculo que não existe em [RRK95] -, valor este que será utilizado no destino para a exibição de n_e. Este procedimento é descrito a seguir. 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, $p^-(n)$ indica o menor tempo de exibição de uma unidade de mídia n.

Sendo $t_g(n)$ o tempo da geração da unidade de mídia n e $\tau_g(n)$ o tempo da chegada ao SS da unidade de mídia n, o SS determina o menor e o maior tempos da geração das unidades de mídia n_m e n_e, utilizando os tempos da chegada ao SS de n_m e n_e e os atrasos Δ_{min} e Δ_{max} para n_m e n_e, da seguinte maneira:

$$t_g^-(n_m) = \tau_g(n_m) - \Delta^m_{\max}, \quad (1)$$

$$t_g^+(n_m) = \tau_g(n_m) - \Delta^m_{\min}, \quad (2)$$

$$t_g^-(n_e) = \tau_g(n_e) - \Delta^e_{\max} \text{ e} \quad (3)$$

$$t_g^+(n_e) = \tau_g(n_e) - \Delta^e_{\min}. \quad (4)$$

[3] Na seção 3.3 há o conceito de ETR de exibição.

Os intervalos $[t_g^-(n_m), t_g^+(n_m)]$ e $[t_g^-(n_e), t_g^+(n_e)]$ podem estar superpostos ou não. Caso haja superposição[4] - caso não tratado em [RRK95]:

Sendo $n_m - \upsilon^-$ o menor número de seqüência cuja UM correspondente do mestre poderia ter sido gerada concorrentemente com n_e e $n_m + \upsilon^+$ o maior número de seqüência cuja UM correspondente do mestre poderia ter sido gerada concorrentemente com n_e, para a UM $n_m - \upsilon^-$ ter sido gerada concorrentemente com n_e, de acordo com a figura 2, a seguinte inequação deve ser satisfeita:

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

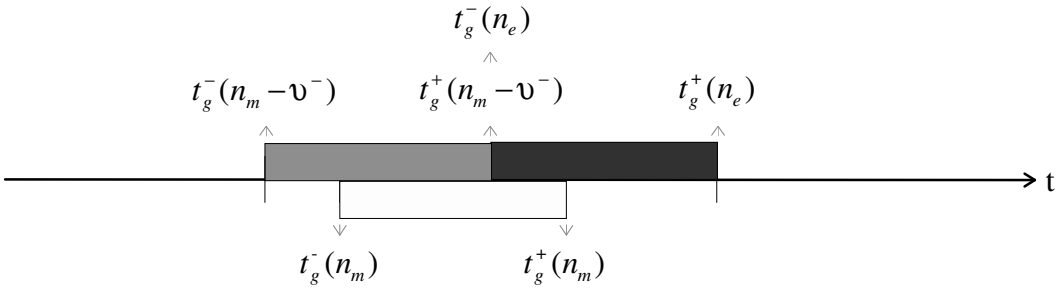

Figura 2: Diagrama temporal para o cálculo de υ^-.

Já para a UM $n_m + \upsilon^+$ ter sido gerada concorrentemente com n_e, a seguinte inequação deve ser satisfeita:

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

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

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

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

e utilizando-se (5), (6), (7) e (8), como υ^- e υ^+ devem ser inteiros[5], diferentemente de [RRK95], tem-se:

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

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

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

[4] Os cálculos das ETRs de n_e no caso de não haver superposição são feitos de maneira análoga.

[5] Os símbolos $\lfloor a \rfloor$ e $\lceil a \rceil$ significam, respectivamente, o maior inteiro ≤ a e o menor inteiro ≥ a.

Definindo-se $\delta t_{g_{\max}}(n_m, n_e) = \max(|t_g^+(n_m) - t_g^-(n_e)|, |t_g^+(n_e) - t_g^-(n_m)|)$, para escolher o melhor valor para a ETR de n_e, calcula-se $\delta t_{g_{\max}}(n_m + \upsilon, n_e)$, $\forall \upsilon = [-\upsilon^-, \upsilon^+]$. O valor de υ escolhido será aquele que faça com que $\delta t_{g_{\max}}(n_m + \upsilon, n_e)$ seja o menor. Este cálculo não está presente em [RRK95].

3.2 Continuidade

Para garantir a continuidade da mídia mestra, de acordo com [RaRa93], utiliza-se um pré-armazenamento de UMs e unidades de realimentação (URs) que são transmitidas pelo aparelho de mídia mestra quando da exibição de certas UMs. Uma UR contém a ETR da UM correspondente. Um flag binário, setado pelo SS antes do começo da exibição, é utilizado para saber se o aparelho de mídia mestra deve transmitir uma UR para a UM exibida. O cálculo da menor taxa de envio de URs, garantindo-se a continuidade da mídia mestra é feito a seguir.

Um pré-armazenamento de P UMs antes de começar a exibição no aparelho de mídia mestra é necessário se a seguinte condição **não** for satisfeita, supondo duas UMs enviadas ao mesmo tempo, a fim de evitar ociosidade de *buffer* (*starvation*):

$$\theta \times (1 - \rho) \geq \Delta^m_{\max} - \Delta^m_{\min}. \tag{11}$$

Isto acontece porque a separação máxima entre a primeira e a segunda UMs é $\Delta^m_{\max} - \Delta^m_{\min}$, logo, se esta diferença é maior que $\theta \times (1 - \rho)$ - menor tempo de exibição da primeira UM -, poderia haver ociosidade de *buffer*. O valor de P é dado por [RaRa93]:

$$P = \left\lceil \frac{\Delta^m_{\max} - \Delta^m_{\min}}{\theta \times (1 - \rho)} \right\rceil. \tag{12}$$

Sendo $\tau(n)$ o tempo quando a unidade de mídia n é transmitida pelo SS, c(n) o tempo da chegada ao aparelho de mídia mestra da unidade de mídia n e $\tau(r_n)$ o tempo quando a unidade de realimentação r_n é transmitida pelo destino - início da exibição da UM n -, a exibição da UM n é dada por:

$$p^-(n) = c(r_n) - \Delta^r_{\max} \text{ e} \tag{13}$$

$$p^+(n) = c(r_n) - \Delta^r_{\min}. \tag{14}$$

Sendo n' uma UM transmitida após a recepção da UR r_n, supondo que as UMs são exibidas continuamente, tem-se:

$$p^-(n') = p^-(n) + (n' - n) \times \theta \times (1 - \rho) \text{ e} \tag{15}$$

$$p^+(n') = p^+(n) + (n' - n) \times \theta \times (1 + \rho). \tag{16}$$

Para evitar transbordo de *buffer*, o menor tempo da chegada de n' à mestra não pode ultrapassar o maior tempo de exibição de n'–B, pois o *buffer* é de tamanho B. Calcula-se, então, $\tau^-(n')$:

$$c^-(n') \geq p^+(n'-B) \Rightarrow \tau^-(n') + \Delta^m_{\min} \geq p^+(n'-B),$$
$$\tau^-(n') \geq p^+(n) + (n'-B-n) \times \theta \times (1+\rho) - \Delta^m_{\min}.$$

Já para evitar ociosidade de *buffer*, o maior tempo da chegada de n' à mestra não pode ultrapassar o menor tempo de exibição de n'. Calcula-se, então, $\tau^+(n')$:

$$c^+(n') \leq p^-(n') \Rightarrow \tau^+(n') + \Delta^m_{\max} \leq p^-(n'),$$
$$\tau^+(n') \leq p^-(n) + (n'-n) \times \theta \times (1-\rho) - \Delta^m_{\max}.$$

Logo, o intervalo de transmissão é:

$$\tau^+(n') - \tau^-(n') = B \times \theta \times (1+\rho) - 2 \times \theta \times \rho \times (n'-n) - (p^+(n) - p^-(n)) - (\Delta^m_{\max} - \Delta^m_{\min}). \quad (17)$$

Quanto maior a diferença entre n e n' (UR menos freqüente) menor o intervalo de transmissão. Chega um ponto que o intervalo fica negativo, quando o SS deve parar as transmissões até receber outra UR. O valor máximo para que o intervalo seja não negativo, como

$$p^+(n) - p^-(n) = c(r_n) - \Delta^r_{\min} - c(r_n) + \Delta^r_{\max} = \Delta^r_{\max} - \Delta^r_{\min}, \quad (18)$$

é:

$$n'_{\max} = \left\lfloor n + \frac{B \times \theta \times (1+\rho) - (\Delta^r_{\max} - \Delta^r_{\min}) - (\Delta^m_{\max} - \Delta^m_{\min})}{2 \times \theta \times \rho} \right\rfloor. \quad (19)$$

Como $\tau^+(n'_{\max})$ é o limite para a recepção da próxima UR r e

$$\tau^+(n'_{\max}) = p^-(n'_{\max}) - \Delta^m_{\max},$$
$$\tau^+(r) = p^-(n'_{\max}) - \Delta^m_{\max} - \Delta^r_{\max},$$

se a transmissão da UR r deve ser concorrente com o início da exibição de uma UM n_{max},

$$p^+(n_{\max}) \leq p^-(n'_{\max}) - \Delta^m_{\max} - \Delta^r_{\max},$$
$$p^+(n) + (n_{\max} - n) \times \theta \times (1+\rho) \leq p^-(n) + (n'_{\max} - n) \times \theta \times (1-\rho) - \Delta^m_{\max} - \Delta^r_{\max}.$$

Utilizando-se (18) e (19),

$$n_{\max} - n \leq \frac{\frac{(B \times \theta \times (1+\rho) - (\Delta^r_{\max} - \Delta^r_{\min}) - (\Delta^m_{\max} - \Delta^m_{\min}))}{2 \times \theta \times \rho} \times \theta \times (1-\rho)}{\theta \times (1+\rho)} + \frac{-2 \times \Delta^r_{\max} + \Delta^r_{\min} - \Delta^m_{\max}}{\theta \times (1+\rho)}. \quad (20)$$

Sendo,

$$F = \frac{1}{n_{\max} - n} \Rightarrow \text{menor taxa de envio de URs}$$

$$F = \frac{2 \times \theta \times \rho \times (1+\rho)}{B \times \theta \times (1-\rho^2) - \Delta^r_{\max} \times (1+3\times\rho) + (\Delta^r_{\min} - \Delta^m_{\max}) \times (1+\rho) + \Delta^m_{\min} \times (1-\rho)}. \quad (21)$$

3.3 Exibição Síncrona

Para que a sincronização intermídia seja mantida, de acordo com [RRK95], é feito um novo cálculo do intervalo da ETR de cada UM do escravo, utilizando URs transmitidas concorrentemente com o início da exibição das UMs.

Definição: A ETR de exibição de uma unidade de mídia n_e é o número de seqüência da unidade de mídia do mestre que poderia ter sido exibida ao mesmo tempo que n_e.

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, de maneira análoga à vista na seção 3.1, utilizando-se os atrasos de realimentação, é feito o cálculo do intervalo de ETRs de exibição de n_e. Este cálculo não será apresentado neste artigo.

A assincronia pode existir (mas não ser detectável) mesmo se os intervalos das ETRs de geração e exibição forem iguais. A assincronia é garantida existir somente quando os dois intervalos não se superpõem. A ressincronização pode ser forçada se há certeza ou mesmo uma pequena chance detectável de assincronia; daí surgem duas 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 das ETRs de geração e exibição, respectivamente, da UM n_e. Sendo $\delta e_{\max}(n_e) = \max(|e_g^+(n_e) - e_e^-(n_e)|, |e_g^-(n_e) - e_e^+(n_e)|)$ - isto equivale à máxima assincronia possível do escravo em relação ao mestre, no tempo de exibição de n_e -, tem-se:

- A assincronia do escravo 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 do escravo excede A, se e somente se $e_g^-(n_e) - e_e^+(n_e) > A$ ou $e_e^-(n_e) - e_g^+(n_e) > A$.

3.3.1 Política Conservativa

A política conservativa atua quando a assincronia excede A. Se o escravo está atrasado em relação ao mestre, o número de UMs apagadas é $e_e^-(n_e) - e_g^+(n_e)$ (figura 3). Quando o escravo está adiantado em relação ao mestre, o número de vezes que uma UM será repetida é $e_g^-(n_e) - e_e^+(n_e)$.

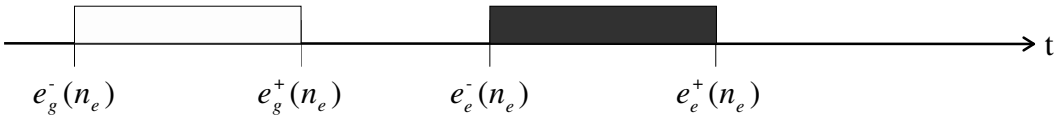

Figura 3: Diagrama temporal mostrando o escravo atrasado em relação ao mestre.

3.3.2 Política Agressiva

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

3.3.3 Outras políticas

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 sua performance quando a assincronia A é grande. A política conservativa trabalha melhor em pequenos níveis de assincronia A e não é boa para grandes níveis de assincronia A.

Várias outras políticas de sincronização podem existir. Rangan et al. [RRK95] destacam duas outras políticas. A política prognosticada explora a periodicidade da gravação e da exibição de mídia para determinar valores para ETRs com maior acurácia e detectar e corrigir assincronias na exibição com maior precisão. A política probabilística utiliza distribuições estatísticas de atrasos de rede (Δ). Apesar de sua boa performance para sincronizar mídias para quaisquer valores de A, os cálculos necessários para a sincronização são complexos.

4 Primitivas de Serviço

O serviço de continuidade e sincronização intermídia está especificado em duas subcamadas: continuidade e sincronização intermídia.

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

A subcamada sincronização intermídia utiliza a subcamada continuidade para prover para a aplicação, pares de mídias (mestra, escrava), cuja assincronia não ultrapassa um valor predeterminado, de cada feixe de sincronização.

A tabela 1 mostra as primitivas de serviço com uma breve descrição e os parâmetros de cada uma.

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
S-CONNECT.*Response*/ S-CONNECT.*Confirm*	*Resposta*/*Confirmação ao*/*do 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	Dados do usuário
S-FEEDBACKED-DATA.*Request*	*Pedido*/*Indicação de pedido* de envio de unidade de realimentação	Dados de realimentação
S-JOIN.*Request*/ S-JOIN.*Indication*	*Pedido*/*Indicação de pedido* de adesão de um fluxo a um feixe de sincronização	Identificador do fluxo Feixe Qualidade de serviço
S-JOIN.*Confirm*	*Confirmação do pedido* de adesão a um feixe de sincronização	Resposta Qualidade de serviço
S-LEAVE.*Request*	*Pedido* de desligamento de um fluxo de um feixe de sincronização	Identificador do fluxo
S-CHANGE-MASTER.*Request*	*Pedido* de mudança de mestre de um feixe de sincronização	Identificador do novo mestre
S-CHANGE-MASTER.*Confirm*	*Confirmação de pedido* de mudança de mestre de um feixe de sincronização	Resposta
S-PAUSE.*Request*	*Pedido* de parada de exibição	---
S-RESUME.*Request*	*Pedido* de retomada de exibição	---
S-FORWARD.*Request*	*Pedido* de avanço de exibição	Velocidade
S-BACKWARD.*Request*	*Pedido* de retrocesso de exibição	Velocidade
S-ABORT.*Indication*	*Indicação de falta de recursos* para processar um feixe	Código do erro

5 Protocolo de Continuidade e Sincronização Intermídia

O cabeçalho das PDUs (*Protocol Data Units*) deve ter tamanho fixo. Os campos do cabeçalho (também de tamanho fixo) devem ser alinhados por quatro octetos, de modo a aproveitar o potencial das máquinas de 32 bites, bem como para acelerar a decodificação dos campos das PDUs [Nune95]. Todas as PDUs deste protocolo de continuidade e sincronização intermídia utilizam o mesmo cabeçalho, de tamanho fixo, com três campos de 32 bites.

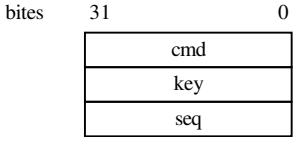

Figura 4: Cabeçalho das PDUs.

O campo *cmd* (*command*) contém informações sobre o tipo da PDU (tipo), o número de octetos necessários para completar o campo de dados do usuário, de modo a tornar seu comprimento múltiplo de 32 bites (*padding*) e as opções de configuração do protocolo (opções). Os bites de 18 a 31 estão reservados para futura utilização.

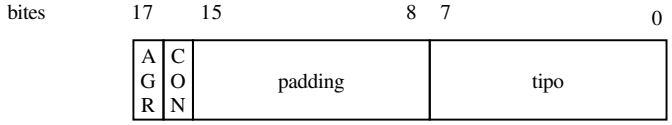

Figura 5: Campo command.

Em opções, CON habilitado em PDUs JR indica utilização da política conservativa e AGR habilitado em PDUs JR indica utilização da política agressiva.

O campo *key* serve para associar uma PDU a uma determinada conexão. Este campo é preenchido pelo SS que é o gerente de conexões de sincronização.

O campo *seq* indica o número de seqüência de PDUs DT e SD. Este campo deve conter o valor zero em quaisquer outras PDUs.

5.1 PDUs do protocolo

As PDUs podem ser divididas em grupos como: conexão, transferência de dados, pré-armazenamento, feixe de sincronização, mudança de mestre, aborto de exibição e funções de videocassete. Cada campo do segmento de dados destas PDUs possui 32 bites.

5.1.1 PDUs de conexão

As PDUs ***CR*** (Connect Request) e ***NCR*** (Numbered Connect Request) são utilizadas para pedido de conexão ponto-a-ponto. A PDU ***NCR*** indica no campo key do cabeçalho qual o número de conexão dado pelo SS. O SS é um gerente de conexões de sincronização. Na PDU ***CR*** o campo key contém o valor zero. O segmento de dados de ambas é igual. Os campos *end_dest* e *end_fonte* contém, respectivamente, os identificadores lógicos das entidades respondedora e iniciadora da conexão.

A PDU ***NCC*** (Numbered Connect Confirm) é utilizada para confirmação de pedido de conexão ponto-a-ponto. O campo *end_resp* contém o identificador lógico da entidade respondedora da conexão. O campo *resposta* contém a resposta (aceitação ou recusa).

5.1.2 PDUs de transferência de dados

A PDU ***DT*** (Data) é utilizada para a transmissão de uma UM. Seu segmento de dados contém um campo *dados* que inclui os octetos de enchimento que venham a ser necessários.

A PDU ***SD*** (Stamped Data) é utilizada para a transmissão de uma UM, após sua passagem pelo SS. Seu segmento de dados contém os campos ETR, UR e dados. O campo *ETR* contém a Estampa de Tempo Relativo da unidade de mídia. O campo *UR* habilitado,

indica que quando esta unidade de mídia estiver sendo exibida, deve ser transmitida uma unidade de realimentação para o SS. Ambos são preenchidos pelo SS. O campo de *dados* é igual ao apresentado na PDU DT.

A PDU ***FD*** (Feedbacked Data) é utilizada para a transmissão de unidades de realimentação. Seu segmento de dados contém o campo *ETR* já apresentado na PDU SD.

5.1.3 PDU de pré-armazenamento

A PDU ***PF*** (Prefetch) é utilizada para pedido de pré-armazenamento de UMs de um mestre no destino. Seu segmento de dados contém os campos ident e num_Um. O campo *ident* contém o identificador do fluxo mestre e o campo *num_UM* contém o número de UMs que devem ser pré-armazenadas.

5.1.4 PDUs de feixe de sincronização

A PDU ***JR*** (Join Request) é utilizada para pedido de adesão a um feixe de sincronização. Seu segmento de dados contém os campos ident, feixe, assinc, delta_min, delta_max, delta_r_min, delta_r_max, theta, ro e *buffer*. O campo *ident* contém o identificador do fluxo e o campo *feixe* contém o identificador do feixe de sincronização. O campo *assinc* contém o valor máximo de assincronia permissível entre o escravo e o mestre, dentro de um feixe de sincronização. Este campo contém o valor zero, no caso do fluxo que está sendo adicionado ao feixe de sincronização ser mestre. Os campos *delta_min*, *delta_max*, *delta_r_min*, *delta_r_max*, *theta* e *ro* foram descritos na seção 3. O campo *buffer* indica qual o tamanho do *buffer* que deve ser alocado no destino para a recepção das UMs do fluxo em questão (seção 3.2).

A PDU ***JC*** (Join Confirm) é utilizada para confirmação de pedido de adesão de um fluxo a um feixe de sincronização. Seu segmento de dados contém os campos *resposta* (aceitação ou recusa), *theta*, *ro* e *buffer*. Os três últimos campos, apresentados na PDU JR, são utilizados quando o SS deseja fazer uma contraproposta.

A PDU ***LR*** (Leave Request) é utilizada para pedido de desligamento de um fluxo de um feixe de sincronização. Seu segmento de dados contém o campo *ident* que é o identificador do fluxo.

5.1.5 PDUs de mudança de mestre

A PDU ***CMR*** (Change Master Request) é utilizada para mudar o mestre de um feixe de sincronização durante a sincronização. Seu segmento de dados contém o campo *ident* que é o identificador do novo mestre.

A PDU ***CMC*** (Change Master Confirm) é utilizada para confirmação de mudança do mestre de um feixe de sincronização durante a sincronização. Seu segmento de dados contém o campo *resposta* (aceitação ou recusa).

5.1.6 PDU de aborto de exibição

A PDU ***AI*** (Abort Indication) é utilizada para indicação de aborto de feixe de sincronização. Seu segmento de dados contém o campo *cod_erro* que indica o código do erro ocorrido.

5.1.7 PDUs de funções de videocassete

As PDUs ***PR*** (Pause Request) e ***RR*** (Resume Request) são utilizadas, respectivamente, para pedido de parada e retomada de exibição. Estas PDUs não contém segmentos de dados.

As PDUs ***FR*** (Forward Request) e ***BR*** (Backward Request) são utilizadas, respectivamente, para pedido de avanço e retrocesso de exibição. Seus segmentos de dados contém o campo *veloc* que indica a velocidade de avanço/retrocesso de exibição.

6 Implementação

O protocolo de continuidade e sincronização intermídia apresentado está sendo implementado no Laboratório do Programa de Engenharia Elétrica da COPPE/UFRJ. A implementação faz uso de um ambiente de desenvolvimento de sistemas de comunicação de alto desempenho com garantias de qualidade de serviço (figura 6), seguindo [BaDu91].

A arquitetura de implementação consiste de três módulos básicos utilizados no desenvolvimento do sistema de comunicação e de alguns módulos específicos, opcionais, que definem as características deste sistema de comunicação. Os módulos básicos são o Módulo de Gerenciamento de Memória (GM), o Módulo de Escalonamento de Tarefas (ET) e o Módulo de Gerenciamento de Temporizações (GT); enquanto que os módulos específicos são o Módulo de Gerenciamento do Sistema, o Sistema de Comunicação e o Módulo de Garantia de Qualidade de Serviço.

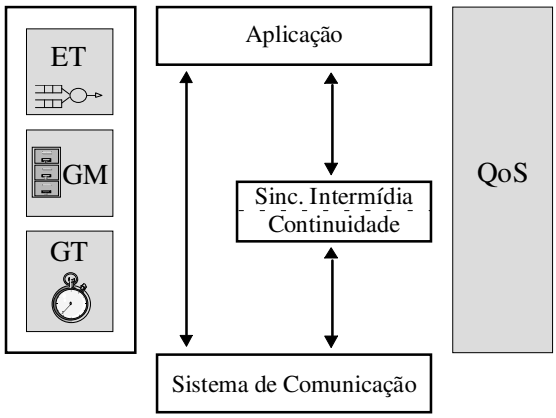

Figura 6: A arquitetura de implementação.

O Módulo de Gerenciamento de Memória divide a memória em regiões de transmissão e recepção, para tornar o seu gerenciamento mais rápido e eficiente. Diferentemente dos gerenciadores de memória de propósito geral, este gerenciador armazena as informações de

controle de alocação separadas dos dados, permitindo uma implementação eficiente e sem cópias de funções de segmentação, remontagem, concatenação, separação, liberação de parte dos dados e outras [Albu95].

O Módulo de Escalonamento de Tarefas armazena e determina a seqüência das tarefas a serem executadas. Além disso, possui funções para permitir níveis de prioridade para os fluxos em tempo real [Albu95]. O comportamento da fila de tarefas é modelado como o de duas filas onde um elemento da fila de menor prioridade (prioridade 2) só será servido quando a fila de maior prioridade estiver vazia (prioridades 1 e 0). Implementando o modelo de duas filas com prioridades através de uma única fila com três pontos de entrada e um único ponto de saída evita-se testes a cada retirada da fila, a fim de verificar se a fila de maior prioridade está vazia ou não [Albu95].

O Módulo de Gerenciamento de Temporizações simula uma analogia a um relógio real através de um contador em módulo. Esta implementação privilegia o desempenho, evitando testes desnecessários a cada interrupção, permitindo portanto a criação dinâmica de um grande número de temporizadores.

O Sistema de Comunicação utiliza os protocolos UDP/IP, de modo a permitir uma comunicação entre estações que não estejam numa mesma rede local. A escolha do UDP deve-se à grande utilização de áudio em aplicações multimídias e ao fato de não ser usual a utilização de retransmissões e reconhecimentos para fluxos isócronos.

O ambiente de desenvolvimento compreende estações de trabalho RISC ligadas por uma rede Ethernet. As estações envolvidas na aplicação possuem placa e câmera para aquisição de vídeo e um sistema para captura e reprodução de áudio. A placa utilizada para a aquisição de vídeo é a SunVideo [Sun94a]. Esta placa permite a captura de até 30 quadros/s e compressão por *hardware* nos formatos MPEG-1, JPEG, CellB e UYVY. A expansão de dados de vídeo é feita por *software*. Um *kit* de desenvolvimento de aplicações utilizando a placa SunVideo (*Software Developer's Kit* - SDK) também está sendo utilizado. Uma biblioteca chamada XIL *Imaging Library*, com funções para a captura, a compressão e o escalamento de quadros, dá suporte ao desenvolvimento da aplicação [Sun94b]. O suporte ao áudio é dado pela estação de trabalho RISC. A compressão de áudio utilizada é a lei μ. A linguagem de programação utilizada é o C e o sistema operacional Solaris.

7 Considerações Finais

Este trabalho apresentou a definição de um serviço de continuidade e de sincronização intermídia. O serviço é apropriado para aplicações multimídias que envolvam fontes de dados armazenados e/ou ao vivo, relações temporais implícitas, múltiplas fontes e destinos e não utilizem relógio global. O serviço provê a sincronização intramídia para as mídias mestras de todos os feixes de sincronização e a sincronização intermídia, para todos os feixes de sincronização, entre os pares (mestra, escrava) para todas as escravas de um feixe de sincronização. Foram definidas primitivas de serviço e tipos e formatos para as unidades de dados de protocolo.

Uma aplicação de teste envolvendo a transferência bidirecional de áudio e vídeo entre duas estações de trabalho está sendo realizada. Encontram-se em desenvolvimento interfaces gráficas com o usuário usando Tcl/Tk, assim como um protocolo de transporte multidestinatário e um escalonador de tarefas com prioridade.

8 Referências

[Albu95] C. V. Albuquerque, "Implementação de Mecanismos de Policiamento para Redes de Alta Velocidade", Tese de Mestrado do Grupo de Teleinformática e Automação da COPPE/UFRJ, 1995.

[AnHo91] D. P. Anderson e G. Homsy, "A Continuous Media I/O Server and Its Synchronization Mechanism", IEEE Computer, pp. 51-57, outubro de 1991.

[BaDu91] L. F. Baginski e O. C. M. B. Duarte, "Um Modelo de Implementação de Alto Desempenho para Sistemas Abertos", IX Congresso da Sociedade Brasileira de Telecomunicações, São Paulo, Brasil, pp. 1941-1945, setembro de 1991.

[CoCa93] J. P. Courtiat e L. F. Rust da Costa Carmo, "A new mechanism for achieving inter-stream synchronization in multimedia communications systems", Relatório Técnico do LAAS, França, novembro de 1993.

[EFI94] L. Ehley, B. Furht e M. Ilyas, "Evaluation of Multimedia Synchronization Techniques", Proceedings of the International Conference on Multimedia Computing and Systems, pp. 514-519, 14 a 19 de maio de 1994.

[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.

[Esco95] J. Escobar, "Multimedia Synchronization over Wide Area Networks", Proceedings of HPN'95, Palma de Mallorca, Espanha, setembro de 1995.

[Ferr92] D. Ferrari, "Delay jitter control scheme for packet-switching internetworks", Computer Communications vol. 15, no. 6, pp. 367-373, julho/agosto de 1992.

[LiDu96] R. M. de Lima e O. C. M. B. Duarte, "Comunicação Sincronizada de Dados Multimídias Pré-Orquestrados", Relatório Técnico do Grupo de Teleinformática e Automação da COPPE/UFRJ, janeiro 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.

[Nico90] C. Nicolaou, "An Architecture for Real-Time Multimedia Communication Systems", IEEE Journal on Selected Areas in Communications vol. 8, no. 3, pp. 391-400, abril de 1990.

[Nune95] M. D. Nunes, "Protocolo Multidestinatário com Configuração de Facilidades e Gerenciamento de Acesso", Tese de Mestrado do Grupo de Teleinformática e Automação da COPPE/UFRJ, 1995.

[RaRa93] S. Ramanathan e P. Venkat 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. Venkat 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.

[SHGC92] D. Shepherd, D. Hutchinson, F. Garcia e G. Coulson, “Protocol support for distributed multimedia applications”, Computer Communications vol. 15, no. 6, pp. 359-366, julho/agosto de 1992.

[SoAg94] S. H. Son e N. Agarwal, “A Model for Specification and Communication of Data for Distributed Multimedia Systems”, Relatório Técnico do Departamento de Ciência da Computação da Universidade de Virgínia, EUA, 1994.

[STCN92] L. F. G. Soares, L. Tucherman, M. A. Casanova e P. R. R. L. Nunes, “Fundamentos de Sistemas Multimídia”, VIII Escola de Computação, Gramado, Brasil, pp. 80-85, 1992.

[Stei90] R. Steinmetz, “Synchronization Properties in Multimedia Systems”, IEEE Journal on Selected Areas in Communications vol. 8, no. 3, pp. 401-412, abril de 1990.

[Sun94a] Sun Microsystems Inc., “SunVideo User’s Guide”, 1994.

[Sun94b] Sun Microsystems Inc., “XIL Programmer’s Guide”, 1994.