Implementação de um Serviço de Sincronização Baseado em Relações Temporais Implícitas[1]

Marcelo Gonçalves Rubinstein
rubi@gta.ufrj.br

Otto Carlos Muniz B. 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

Este trabalho descreve a implementação de um serviço de continuidade e sincronização intermídia e do seu respectivo protocolo utilizando uma arquitetura de alto desempenho. Este serviço visa aplicações multimídias que envolvem fontes de dados armazenados ou ao vivo, relações temporais implícitas, múltiplas fontes e destinos e não utilizam relógio global. O serviço é definido através das suas primitivas de serviço. O tipo e o formato das unidades de dados de protocolo, diagramas em SDL (*Specification and Description Language*) relativos ao protocolo e o ambiente de desenvolvimento são apresentados. Foi utilizada uma arquitetura de implementação de alto desempenho, contendo módulos de escalonamento de tarefas com prioridades, gerenciamento de memória otimizado e gerenciamento de temporizações.

Abstract

This paper defines a continuity and intermedia synchronization service, and specifies its protocol using a high performance architecture. This service aims multimedia applications that involve stored or live data sources, implicit temporal relationships, multiple sources and destinations, and do not utilize global clock. The service is defined by its service primitives. Protocol data units' type and format, SDL diagrams related to protocol, and development environment are presented. A high performance implementation architecture has been used, defining specialized modules of priority-based task scheduling, optimized memory management, and timer management.

---

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

1 Introdução

O novo conjunto de aplicações distribuídas em redes a ser suportado pelas redes de alta velocidade inclui aplicações multimídias interativas com vários participantes, visualização científica, distribuição domiciliar de TV de alta definição, videotelefonia, teleducação, realidade virtual, trabalho cooperativo, transferência de dados a alta velocidade e outras [Albu95]. Para este conjunto de aplicações, novos 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 de 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 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, precisam ser mantidas.

A sincronização pode ser classificada em relação ao fluxo envolvido. A **sincronização intramídia**, também chamada continuidade, corresponde à relação temporal existente entre um dado e o próximo dado de um mesmo fluxo, sendo este fluxo isócrono ou não. A **sincronização intermídia** corresponde à relação temporal entre dados, de dois ou mais fluxos, isócronos ou não. Outra classificação é feita em relação à forma de expressar as relações temporais. A **sincronização contínua**, também chamada sincronização natural ou sincronização ao vivo, mantém relações temporais implícitas de modo contínuo, para uma ou mais mídias. A **sincronização sintética**, também chamada sincronização baseada em eventos, serve-se de relações temporais explícitas entre objetos multimídias. 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, de modo a reforçar o poder da sincronização temporal [CaCo96]. 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. Nenhuma das técnicas garante todos os requisitos das várias aplicações

existentes e estas técnicas podem ser analisadas através de suas principais características [Rubi96, BlSt96].

Este trabalho descreve a implementação de um serviço de sincronização, especificado através de suas primitivas de serviço, apropriado para aplicações multimídias que envolvem fontes de dados armazenados ou ao vivo, relações temporais implícitas, múltiplas fontes e destinos e não utilizam relógio global. O serviço provê a sincronização intramídia para as mídias mestras de todos os feixes de sincronização (conjunto de fluxos que possuem relações temporais entre si) e a sincronização intermídia, para todos os feixes de sincronização. A fase de transferência de dados é especificada através de diagramas em SDL [ITU-T94]. Uma arquitetura de implementação de alto desempenho e o ambiente de desenvolvimento são apresentados.

O trabalho está organizado da seguinte maneira: a seção 2 apresenta o mecanismo de sincronização, a seção 3 descreve o serviço de sincronização proposto, a seção 4 descreve a implementação do protocolo e na seção 5 são apresentadas as conclusões.

2 Mecanismo de sincronização

O mecanismo de sincronização destina-se às aplicações que envolvem fontes de dados armazenados ou ao vivo, relações temporais implícitas, múltiplas fontes e destinos e não utilizam relógio global.

Para ter-se uma única base de tempo (essencial em sincronização), uma arquitetura centralizada foi 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 arquitetura centralizada é constituída de um Servidor de Sincronização (SS) e de diversas estações multimídias interconectadas por uma rede de computadores. A idéia chave da centralização é concentrar a maior carga de processamento no Servidor de Sincronização, permitindo a utilização de estações multimídias de menor capacidade de processamento e, portanto, de menor custo. Além disto, o Servidor de Sincronização permite o atendimento de diversas aplicações ao mesmo tempo, garantindo que dados gerados por fontes distribuídas sejam apresentados de forma sincronizada nos seus destinos.

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 elementar de uma mídia, de um feixe de sincronização, a chegar ao SS. A unidade elementar de uma mídia será doravante denominada **Unidade de Mídia** (UM). O Servidor de Sincronização

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 *buffers* nos destinos fiquem vazios (*starvation*) ou cheios (*overrun*).

A técnica de sincronização utilizada, descrita com maiores detalhes em [RuDu96, 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 Δ_{min} e Δ_{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, de modo a evitar que *buffers* nos destinos fiquem vazios ou cheios. 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 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 *n*, $\theta(n)$, 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. 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 de mídia, enquanto que as **repetições** correspondem à 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.

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 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.

3 Camada Sincronização

3.1 Serviço de sincronização

O presente trabalho concentra as funções de sincronização em uma camada denominada Sincronização. O serviço de sincronização está especificado em duas subcamadas: continuidade (sincronização intramídia) e sincronização intermídia.

Devido às características centralizadas atribuídas ao Servidor de Sincronização, as funções da camada Sincronização são assimétricas. Portanto, as Máquinas de Estados do SS e das outras estações são distintas. A arquitetura de protocolos utilizada é apresentada na figura 1.

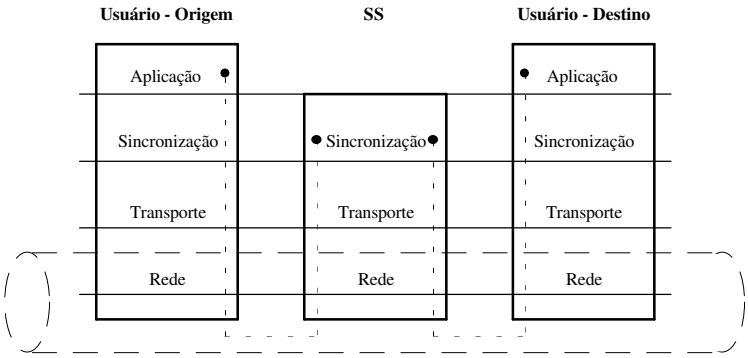

Figura 1: Arquitetura de protocolos utilizada.

Cabe à subcamada continuidade oferecer um 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 pares de mídias (*mestra*, *escrava*), cuja assincronia não ultrapassa um valor predeterminado $A_{\max}$, de cada feixe de

sincronização. Na tabela 1 são apresentadas as primitivas de serviço com uma breve descrição de cada uma.

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

Primitivas	Descrição
S-CONNECT.*Request*/ S-CONNECT.*Indication*	*Pedido*/*Indicação de pedido* de estabelecimento de conexão
S-CONNECT.*Response*/ S-CONNECT.*Confirm*	*Resposta*/*Confirmação a*/*de pedido* de estabelecimento de conexão
S-DISCONNECT.*Request*/ S-DISCONNECT.*Indication*	*Pedido*/*Indicação de pedido* de liberação de conexão
S-DATA.*Request*/ S-DATA.*Indication*	*Pedido*/*Indicação de pedido* de envio de unidade de mídia
S-FEEDBACKED-DATA.*Request*	*Pedido* de envio de unidade de realimentação
S-JOIN.*Request*	*Pedido* de adesão a um feixe
S-JOIN.*Confirm*	*Confirmação de pedido* de adesão a um feixe
S-LEAVE.*Request*	*Pedido* de desligamento de um fluxo de um feixe
S-ABORT.*Indication*	*Indicação de falta de recursos* para processar um feixe

3.2 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. Na tabela 2 são apresentadas as PDUs do protocolo de sincronização com uma breve descrição de suas utilizações.

Tabela 2: Unidades de dados de protocolo de sincronização.

PDUs	Descrição
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
DR (Disconnect Request)	Pedido de desconexão ponto-a-ponto
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
PF (Prefetch)	Pedido de pré-armazenamento de dados
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
ERROR	Fornecimento de diagnósticos para diversas situações de erro

O protocolo possui cinco procedimentos principais: abertura/fechamento de conexão, adesão/desligamento de grupo e transferência de dados.

Na fase de transferência de dados, estes são transmitidos ao SS através de PDUs DT seqüencialmente numeradas. O SS, ao receber uma PDU DT, verifica se esta é relativa a uma mídia mestra ou escrava. No caso da PDU ser relativa a uma mídia mestra, o SS atribui ao campo ETR o número de seqüência da PDU DT - gerando uma PDU SD - e verifica se para esta PDU será necessário o envio de uma UR. No caso da PDU ser relativa a uma mídia escrava, o SS calcula a ETR da UM em questão e envia a PDU SD para o destino. Um diagrama de recepção da PDU DT pelo SS é apresentado na figura 2. A estação remota, ao receber uma PDU SD, verifica o valor da variável booleana *Espera_Pre_Armazenamento*[*feixe*]. Se a variável for falsa, o tempo de exibição da UM é calculado e, ao estourar o temporizador correspondente, uma primitiva S-DATA.Indication é entregue ao usuário. Se a variável for verdadeira, a criação dos temporizadores só começará após o recebimento da P-ésima UM [Rubi96]. Cada temporizador que estourar gerará uma primitiva S-DATA.Indication. Um diagrama de recepção da PDU SD pelo destino é apresentado na figura 3.

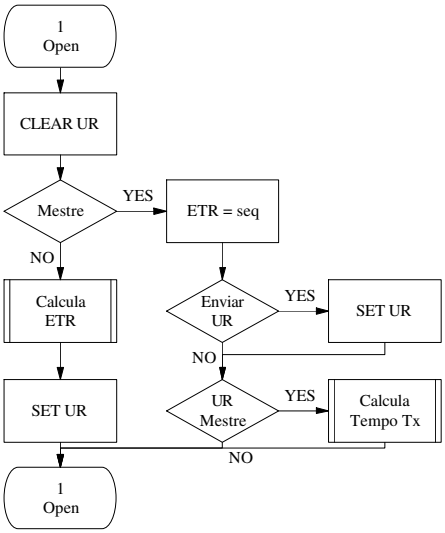

Figura 2: Diagrama em SDL da recepção da PDU DT pelo SS.

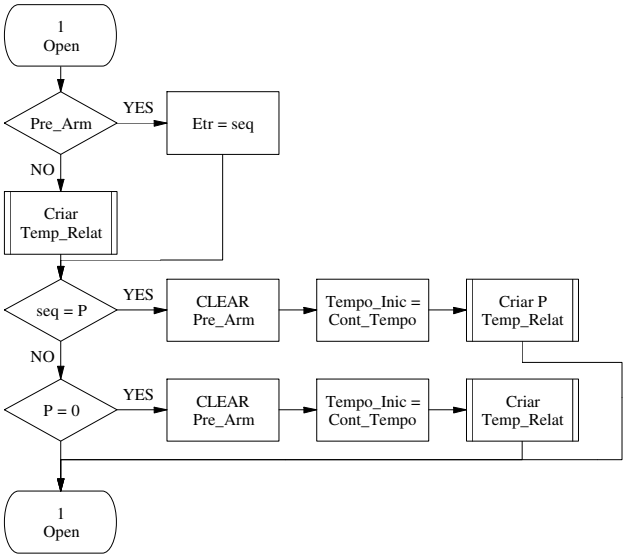

Figura 3: Diagrama em SDL da recepção da PDU SD pelo destino.

4 Implementação

O protocolo apresentado foi implementado utilizando-se um ambiente de desenvolvimento de sistemas de comunicação de alto desempenho com garantias de qualidade de serviço denominado GTA/MM (GTA/MultiMídia) [Albu95]. Por esta arquitetura ser uma estrutura modular e flexível, é possível escolher e configurar os módulos de acordo com as características do sistema desejado, bem como criar novos módulos.

Cada módulo possui diversas funções, com o objetivo de tornar o sistema bastante flexível. No Módulo de Gerenciamento de Memória (GM) são oferecidas funções de alocação e liberação de estruturas de dados inerentes à arquitetura, tais como estruturas de primitivas e de parâmetros, blocos de dados de tamanho fixo e dados de tamanho variável. O Módulo de Escalonamento de Tarefas (ET) possui funções para inserir, executar e retirar tarefas de baixa, média e alta prioridade, para atender aos dados que possuem restrições temporais. O Módulo de Gerenciamento de Temporizações (GT), de grande utilidade em sistemas de comunicação em tempo real, oferece funções para iniciar, criar, interromper, estourar, reiniciar e finalizar temporizadores.

A comunicação entre as camadas é feita de forma assíncrona e as interfaces entre camadas consistem de um par de filas (uma para cada sentido de transmissão) por conexão, exceto a interface SINC_T (Sincronização-Transporte) que não possui a fila SINC_T. Portanto, a comunicação Sincronização-Transporte é feita diretamente, através de chamada de procedimentos. O esquema da arquitetura proposta é apresentado na figura 4.

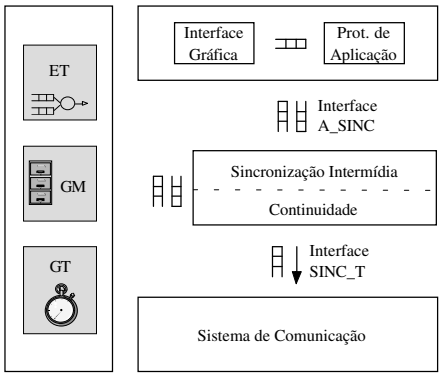

Figura 4: Arquitetura de implementação.

Foram feitas algumas modificações nos módulos de Escalonamento de Tarefas e Gerenciamento de Temporizações da Arquitetura GTA/MM de modo a permitir a implementação das funcionalidades propostas pelo serviço de sincronização e torná-los mais flexíveis [Rubi96].

4.1 Ambiente de desenvolvimento

O ambiente de desenvolvimento compreende estações de trabalho SUN-SPARC, interconectadas através de uma rede Ethernet, com placa e câmera para aquisição de vídeo e um sistema para captura e reprodução de áudio. A linguagem de programação utilizada é o C e o sistema operacional Solaris 2.4.

A placa utilizada para a aquisição de vídeo é a SunVideo. Esta placa permite a captura de até 30 quadros/s e compressão por *hardware* nos formatos de vídeo MPEG-1, JPEG, CellB e UYVY. A descompressão de dados de vídeo é feita por *software*. Um pacote de desenvolvimento de aplicações (*Software Developer's Kit* - SDK) utilizando a placa SunVideo e uma biblioteca chamada XIL *Imaging Library* , contendo funções para a captura, a compressão e o escalamento de quadros, dão suporte ao desenvolvimento das aplicações. Neste trabalho, o formato de vídeo utilizado é o JPEG.

O áudio é capturado ao vivo a uma taxa de 8000 amostras/s. A lei μ, padrão para dados de voz utilizado nos EUA, Canadá e Japão, é utilizada como lei de compressão, reduzindo o tamanho de uma amostra de 12 *bits* para 8 *bits*. Com isto, a taxa de captura de áudio é de 64 kbps.

4.2 Sistema implementado

O sistema implementado divide-se em: programa principal, módulos de áudio e vídeo e camadas Aplicação, Sincronização e Transporte.

A Aplicação está dividida em interface gráfica e protocolo de Aplicação, que se comunicam através de uma fila. Somente o protocolo de Aplicação se comunica com a camada Sincronização, conforme a figura 3. A interface gráfica foi criada utilizando-se a ferramenta de auxílio à confecção de janelas Xview.

O sistema de comunicação utiliza os protocolos UDP/IP, de modo a permitir uma comunicação entre estações. 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 primeiro protótipo deste sistema, constituído de aproximadamente 10000 linhas de código, encontra-se em funcionamento no Laboratório do Grupo de Teleinformática e Automação da COPPE/UFRJ.

5 Conclusões

Este trabalho apresentou um serviço de continuidade e de sincronização intermídia 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. 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. O serviço é definido através das primitivas de serviço e o seu respectivo protocolo é especificado através dos tipos das unidades de dados de protocolo e de diagramas em SDL.

A implementação do protocolo utiliza um ambiente de desenvolvimento de sistemas de comunicação de alto desempenho com garantias de qualidade de serviço denominado Arquitetura GTA/MM. Esta arquitetura contém um módulo de gerenciamento de memória otimizado, um módulo de escalonamento de tarefas com prioridades e um módulo de gerenciamento de temporizações.

O sistema de comunicação foi implementado na linguagem C e roda em SUN-SPARC, com sistema operacional Solaris 2.4, 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. O sistema de comunicação implementado utilizou os protocolos UDP/IP, de modo a permitir uma comunicação entre estações.

A continuação deste trabalho visa o uso do protocolo IPv6 e do protocolo de reserva de recursos RSVP (*resource ReSerVation Protocol*), e também de um protocolo de transporte multidestinatário. Está prevista também a migração de todo o sistema para um ambiente de rede ATM.

Referências

[Albu95] C. V. N. Albuquerque, "Arquitetura de Implementação de Alto Desempenho para Sistemas de Comunicação Multimídia", *Tese de Mestrado*, PEE-COPPE/UFRJ, abril de 1995.

[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.

[ITU-T94] International Telecommunications Union - Technical section, "Functional Specification and Description Language SDL", *ITU-T Recommendation Z.100*, junho 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.

[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.