

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Um Suporte para Aplicações Cooperativas sobre a Internet

Dissertação submetida à Universidade Federal de Santa Catarina
para a obtenção do grau de
Mestre em Engenharia Elétrica

Nelkis de la Orden Medina

Florianópolis, 10 de março de 1997

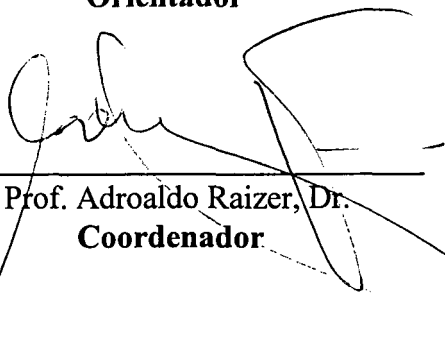
Um Suporte para Aplicações Cooperativas sobre a Internet

Nelkis de la Orden Medina

Esta dissertação foi julgada para a obtenção do título de
Mestre em Engenharia
especialidade **Engenharia Elétrica**,
área de concentração **Controle, Automação e Informática Industrial**,
e aprovada em sua forma final pelo Curso de Pós-Graduação.

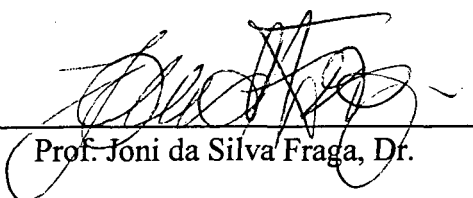


Prof. Jean-Marie Farines, Dr. Ing.
Orientador

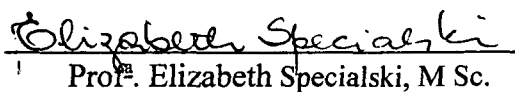


Prof. Adroaldo Raizer, Dr.
Coordenador

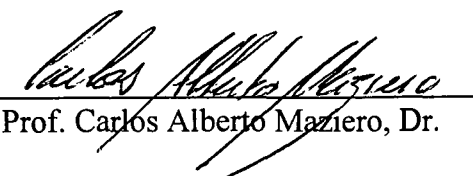
Banca Examinadora:




Prof. Joni da Silva Fraga, Dr.



Prof.^a Elizabeth Specialski, M Sc.



Prof. Carlos Alberto Maziero, Dr.



Roberto Willrich, Dr.

AGRADECIMENTOS

Ao Prof. Dr. Jean-Marie Farines pelo trabalho de orientação e apoio no desenvolvimento deste trabalho.

Aos membros da banca examinadora pelas críticas e comentários.

Aos professores Profs. Eugênio de Bona Castelan, Edson de Pieri e Julio Elias Normey Rico por sua preocupação e ajuda, especialmente no começo desta jornada.

Ao pessoal da CPGEEL, Ênio Valmor Kassick e Wilson Silva Costa por todo o auxílio dado.

Aos colegas do Instituto de Cibernética Matemática e Física, por ter despertado em mim o interesse pela pesquisa.

A todos os colegas do LCMI, especialmente a Udo Fritzke Jr. pelos comentários e sugestões.

Aos amigos Maria Madalena Dias, Nardenio Almeida Martins, Rosamelia Parizotto, Richard Duarte Ribeiro, Georgina Vivanco e Nestor Bercovich.

Aos meus pais Orlando de la Orden Marrero e Zaida Medina Soto pelo exemplo de luta e amor.

A meu esposo e acima de tudo amigo, Damián Rodríguez, pelo amor e carinho compartilhados em todo momento.

Ao meu filho David, por seu carinho.

À UFSC e a CNPq pelo suporte material e financeiro.

ÍNDICE

ÍNDICE DE FIGURAS.....	vii
RESUMO.....	viii
ABSTRACT.....	ix
CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1 APRESENTAÇÃO DO TRABALHO.....	2
CAPÍTULO 2 - O TRABALHO COOPERATIVO	3
2.1 INTRODUÇÃO	3
2.2 CONCEITOS RELACIONADOS COM OS SISTEMAS DE TRABALHO COOPERATIVO OU <i>GROUPWARE</i>	4
2.3 CLASSIFICAÇÃO DOS SISTEMAS DE TRABALHO COOPERATIVO	4
2.3.1 Os Aspectos Temporal e Espacial dos Sistemas de Trabalho Cooperativo.....	5
2.3.2 Classificação Funcional dos Sistemas de Trabalho Cooperativo.....	5
2.4 CARACTERÍSTICAS FUNCIONAIS DOS SISTEMAS DE TRABALHO COOPERATIVO	8
2.5 SUPORTE PARA APLICAÇÕES DE TRABALHO COOPERATIVO	10
2.6 CONCLUSÕES.....	11
CAPÍTULO 3 - MBONE E IP MULTICAST.....	12
3.1 INTRODUÇÃO	12
3.2 GENERALIDADES SOBRE REDES DE COMPUTADORES	13
3.2.1 Interconexão de Redes	13
3.2.2 Mecanismo de Comutação de Pacotes Internet.....	13
3.2.3 As Classes de Endereços Internet	13
3.3 IP MULTICAST	14
3.3.1 Endereçamento Multicast.....	15
3.3.2 Protocolo de Gerenciamento de Grupos Internet.....	16
3.3.3 Configurar o Sistema para Suportar Multicast.....	17
3.4 A REDE MBONE	18
3.4.1 Roteamento Multicast	18
3.4.1.1 Protocolos de Roteamento Multicast.....	18
Distance Vector Multicast Routing Protocol	18
Multicast Open Shortest Path First.....	19
Protocol Independent Multicasting	19
3.4.1.2 Time To Live (TTL).....	20
3.4.1.3 Túneis	22
Túneis Truncados	22
Túneis Pruned.....	22

3.4.2 Tráfego no MBone.....	22
3.4.3 Topologia MBone.....	24
3.5 FERRAMENTAS MULTIMÍDIAS SOBRE MBONE.....	25
3.5.1 Gerenciamento de Sessões.....	26
3.5.2 Vídeo.....	26
3.5.3 Áudio.....	29
3.5.4 Compartilhamento de Documentos.....	30
3.6 MONITORANDO O MBONE.....	31
3.7 LIMITAÇÕES DAS APLICAÇÕES MULTIMÍDIAS COOPERATIVAS SOBRE MBONE.....	36
3.8 CONCLUSÕES.....	37

CAPÍTULO 4 - UMA ABORDAGEM PARA APLICAÇÕES COOPERATIVAS MULTIMÍDIAS SOBRE MBONE..... 38

4.1 INTRODUÇÃO.....	38
4.2 O PROBLEMA DA ORDENAÇÃO NAS APLICAÇÕES DE TRABALHO COOPERATIVO.....	39
4.3 LIMITAÇÕES DO MBONE PARA O DESENVOLVIMENTO DE APLICAÇÕES DE TRABALHO COOPERATIVO.....	39
4.3.1 Ordenação.....	40
4.3.2 Tempo de validade das informações.....	41
4.4 AS SOLUÇÕES AO PROBLEMA DA ORDENAÇÃO COM TEMPO DE VALIDADE DAS MENSAGENS.....	41
4.4.1 Ordenação Causal.....	42
4.4.2 Ordenação Δ -Causal.....	43
4.4.2.1 Princípios e Definição.....	43
4.4.2.2 O Uso da Δ -Causalidade em Aplicações Cooperativas Multimídias.....	45
4.4.2.3 A Necessidade do Tempo Físico para a Ordenação Δ -Causal.....	45
4.5 ARQUITETURA PARA O TRABALHO COOPERATIVO EM REDES INTERNET.....	46
4.6 PROTOCOLO DE ORDENAÇÃO DE MENSAGENS.....	48
4.6.1 Descrição do Protocolo.....	49
4.7 PROTOCOLO DE SINCRONIZAÇÃO DE RELÓGIOS.....	52
4.7.1 Descrição do Network Time Protocol (NTP).....	52
4.7.2 Funcionamento do NTP.....	53
4.7.3 Modos de operação.....	55
4.7.4 O NTP Utilizado para Sincronizar os Relógios dos Participantes de uma Sessão de Trabalho Cooperativo.....	56
4.8 MECANISMO DE CONTROLE DE CONCORRÊNCIA.....	57
4.9 CONCLUSÕES.....	57

CAPÍTULO 5 - UM EXEMPLO DE APLICAÇÃO DE TRABALHO COOPERATIVO MULTIMÍDIA SOBRE MBONE..... 58

5.1 INTRODUÇÃO.....	58
5.2 ARQUITETURA DA APLICAÇÃO COOPERATIVA MULTIMÍDIA.....	58
5.3 IMPLEMENTAÇÃO DO PROTOCOLO DE ORDENAÇÃO DE MENSAGENS.....	60
5.3.1 Variáveis Globais.....	61

5.3.2 A classe Canal.....	61
5.3.3 A classe Emissor de Mensagem.....	62
5.3.4 A classe Receptor de Mensagem	62
5.3.5 A classe Relógio	64
5.4 CONFIGURAÇÃO DO NETWORK TIME PROTOCOL (NTP)	65
5.5 EXPERIMENTOS SOBRE A APLICAÇÃO.....	65
5.5.1 Configuração do NETWORK TIME PROTOCOL (NTP) na rede local	65
5.5.2 Teste do Funcionamento do Protocolo de Ordenação de Mensagens.....	66
5.6 CONCLUSÕES.....	66
CAPÍTULO 6 - CONCLUSÕES E PERSPECTIVAS.....	68
REFERÊNCIAS BIBLIOGRÁFICAS	70

ÍNDICE DE FIGURAS

Fig. 3.1	As classes de endereços Internet.....	14
Fig. 3.2	Formato de endereço multicast classe D	15
Fig. 3.3	Mensagem query no IGMP.....	17
Fig. 3.4	Duas LANs rodando TCP/IP e interconectadas através do roteador A	21
Fig. 3.5	Representação da topologia Mbone.....	25
Fig. 3.6	Janela do SD.....	26
Fig. 3.7	Interface do NetVideo.....	27
Fig. 3.8	Interface do IVS.....	29
Fig. 3.9	Interface do Visual Audio Tool.....	29
Fig. 3.10	Participantes compartilham uma sessão usando o WB	31
Fig. 3.11	Saída do software mtrace para o teste 1	33
Fig. 3.12	Saída do software mtrace para o teste 2	34
Fig. 4.1	Comunicação ordenada causalmente.....	42
Fig. 4.2	Comunicação com ordenação Δ -Causal	44
Fig. 4.3	Arquitetura para desenvolver aplicações de trabalho cooperativo sobre a Internet.....	47
Fig. 4.4	Subrede NTP.....	53
Fig. 5.1	Arquitetura da aplicação cooperativa multimídia em cada sítio participante	59
Fig. 5.2	Relação entre a classe Relógio e o Protocolo de Sincronização de Relógios NTP	64

RESUMO

A contínua evolução tecnológica tem projetado o trabalho cooperativo no mundo das aplicações em redes geograficamente distribuídas (*Wide Area Networks - WANs*), em particular na Internet. Entretanto, este novo campo de aplicação demanda ainda um esforço para a adaptação da tecnologia existente, tendo em vista as funcionalidades requeridas pelas aplicações cooperativas.

A comunicação de grupo é um dos conceitos de base que permite programar aplicações de Trabalho Cooperativo. O conceito de endereçamento *multicast* para a comunicação de grupo na Internet foi adotado em 1992 pela IETF (*Internet Engineering Task Force*) e resultou na proposta da rede MBone (*Multicast Backbone*).

Neste contexto, este trabalho tem como meta principal propor um suporte que garanta o cumprimento dos requisitos que as sessões de trabalho cooperativo exigem, com o objetivo de permitir a construção de aplicações cooperativas sobre Internet. Este suporte permite a ordenação das mensagens levando em consideração o tempo de validade. As principais características desta abordagem são apresentadas e discutidas neste trabalho. A proposta é finalmente implementada e verificada sobre MBone.

ABSTRACT

The continuous evolution of technology allows the introduction of cooperative work applications into the world of WANs (Wide Area Networks), particularly the Internet. However, this new field of applications still demands a great deal of efforts in order to fit the existing technologies to the requirements of cooperative systems.

Group communication is one of the basic concepts for the programming of cooperative work applications. The concept of multicast addressing for group communication on the Internet was adopted in 1992 by the IETF (Internet Engineering Task Force) and resulted in the proposal of the MBone (Multicast Backbone) network.

In this context, the main goal of this work deals with the proposal of a suitable support for cooperative applications on the Internet that allows to order messages taking their lifetimes into account. The main characteristics of the approach are presented and discussed, and the proposal is finally implemented and verified on the MBone.

CAPÍTULO 1

INTRODUÇÃO

O Trabalho Cooperativo por meio de Computador (*Computer Supported Cooperative Work - CSCW*) inclui uma grande variedade de aplicações que vai desde editores cooperativos até sistemas de engenharia concorrente e sistemas de teleconferências. Estas aplicações favorecem a integração dos usuários e permitem um aumento de rendimento na execução das tarefas, em relação ao trabalho individual isolado.

Além do mais, a contínua evolução tecnológica (redes de alta velocidade, WEB, suporte multimídia, etc.) tem projetado o trabalho cooperativo no mundo das aplicações em redes de longa distância, em particular na Internet, oferecendo possibilidades de intercâmbio de idéias e de execução de tarefas em conjunto para um número maior de pessoas.

A comunicação de grupo é um dos conceitos de base que permite programar aplicações de Trabalho Cooperativo. O conceito de endereçamento *multicast* para a comunicação de grupo na Internet [Deering89] foi adotado em 1992 pela *Internet Engineering Task Force (IETF)* e resultou na proposta da rede Mbone (*Multicast Backbone*) [Casner92].

Na rede Mbone, tem sido desenvolvidas diversas aplicações multimídias. No entanto, no caso de aplicações de Trabalho Cooperativo com uma forte necessidade de interatividade, a rede Mbone apresenta limitações referentes à consistência e à coordenação da sessão cooperativa.

Neste contexto, o trabalho aqui desenvolvido foi motivado principalmente pela necessidade de ser proposto um suporte que garanta o cumprimento dos requisitos que as sessões de trabalho cooperativo exigem, com o objetivo de tornar possível o desenvolvimento de aplicações cooperativas multimídias sobre a Internet.

1.1 APRESENTAÇÃO DO TRABALHO

O texto deste trabalho será subdividido em função dos pontos principais que foram desenvolvidos no andamento do projeto.

Assim, o Capítulo 2 apresentará a rede Mbone e descreverá as ferramentas multimídias sobre Mbone e os testes feitos neste trabalho com as mesmas. Serão discutidas as limitações das aplicações multimídias sobre a rede Mbone, principalmente no que diz respeito ao uso em aplicações de Trabalho Cooperativo.

O Capítulo 3 fará uma introdução ao tema Trabalho Cooperativo, onde serão classificadas as aplicações cooperativas e também apresentadas as características funcionais mais importantes destas aplicações. A partir destas características são então citados alguns requisitos gerais necessários para o suporte computacional destes sistemas.

O Capítulo 4 apresentará a proposta de uma plataforma de suporte para o desenvolvimento de aplicações cooperativas síncronas que possam fazer uso de mídias contínuas. Será apresentado o problema da ordenação nas aplicações de trabalho cooperativo. Logo a seguir serão colocados os limites do uso do Mbone para aplicações de Trabalho Cooperativo, envolvendo dados multimídias. Uma solução para a ordenação de mensagens com limitações tempo-real em redes não confiáveis será apresentada. Em seguida será descrita a plataforma proposta para aplicações cooperativas multimídias e a interligação entre os diferentes níveis da arquitetura desta.

No Capítulo 5 tem-se como meta apresentar um exemplo de aplicação de trabalho cooperativo sobre Mbone fazendo uso do suporte proposto. Será apresentada a arquitetura da aplicação e serão detalhados os diferentes elementos que a constituem e a forma em que os mesmos se relacionam.

Por fim, as conclusões do trabalho realizado e as perspectivas de trabalhos futuros serão encontradas no Capítulo 6.

CAPÍTULO 2

O TRABALHO COOPERATIVO

2.1 INTRODUÇÃO

O rápido crescimento das redes interconectadas tem levado ao surgimento de uma infraestrutura de computação com enorme potencial que permite o desenvolvimento de diferentes tipos de aplicações distribuídas. Dentro das mais relevantes estão as orientadas à colaboração entre múltiplos usuários. Neste contexto, a disciplina Trabalho Cooperativo com Suporte Computacional (*Computer Supported Cooperative Work - CSCW*) vem estudando aspectos tecnológicos e sociais do Trabalho Cooperativo com o suporte de recursos computacionais compartilhados[Ellis91].

Junto com a contínua evolução tecnológica (redes de alta velocidade, suporte multimídia, etc.), os produtos derivados desta área estão modificando a forma como pesquisadores, engenheiros, científicos e até gerentes interagem diariamente em suas vidas profissionais. A variedade de aplicações nesta área vão desde editores cooperativos simples e ferramentas de desenho compartilhadas até sistemas de engenharia concorrente e sistemas completos de teleconferências [Cosquer96].

Neste capítulo são apresentados inicialmente alguns conceitos gerais relacionados com os sistemas de Trabalho Cooperativo ou *Groupware*. Também são apresentadas classificações dos sistemas de Trabalho Cooperativo a luz de aspectos temporais e espaciais destes sistemas. Logo a seguir são colocadas as características funcionais de sistemas de Trabalho Cooperativo. Finalmente, a partir destas características são então citados alguns requisitos gerais necessários para o suporte computacional destes sistemas.

2.2 CONCEITOS RELACIONADOS COM OS SISTEMAS DE TRABALHO COOPERATIVO OU *GROUPWARE*

Sistemas de Trabalho Cooperativo ou *Groupware* é a denominação que designa o agrupamento de sistemas que suportam a cooperação de um conjunto de usuários sobre uma tarefa comum. Estes sistemas possuem uma série de vantagens, entre as quais colocaremos as mais importantes. Primeiro, mais que estender as fronteiras da sala de reunião, eles permitem às pessoas acessar e compartilhar recursos, integrando e possivelmente exportando elementos a partir dos seus ambientes computacionais locais. Segundo, eles permitem uma boa mistura de operações síncronas e assíncronas que ajudam a alcançar objetivos que às vezes são muito difíceis de conseguir em reuniões “face-a-face”.

Organizar uma atividade cooperativa baseada em computadores, também conhecida como sessão, implica a existência de mecanismos para programar os eventos, que são conhecidos como *gerenciadores de sessões*¹. O termo *ciclo de vida de uma sessão* se refere a todas as etapas de uma sessão. Uma sessão pode estar em uma das seguintes etapas: *etapa de registro*, *etapa de anúncio* e finalmente na *etapa ativa* quando a cooperação está realmente sendo realizada.

Denomina-se *participantes*, aqueles usuários que interagem durante a *sessão*. O *ambiente de trabalho compartilhado* é, por definição, o conjunto de ferramentas envolvidas na cooperação, as quais normalmente atuam sobre objetos gráficos e texto; no entanto cada vez é mais comum a incorporação de canais multimídias que permitem a troca de informação de áudio e imagens [Cosquer96].

2.3 CLASSIFICAÇÃO DOS SISTEMAS DE TRABALHO COOPERATIVO

Os sistemas de Trabalho Cooperativo são classificados a partir de características como a funcionalidade e os contextos geográficos e temporais nos quais os participantes cooperam, conforme será visto a seguir.

¹ No capítulo 3 é descrita uma ferramenta para o gerenciamento de sessões no Mbone, o *Session Directory* (SD).

2.3.1 Os Aspectos Temporal e Espacial dos Sistemas de Trabalho Cooperativo

As aplicações de Trabalho Cooperativo vem sendo analisadas segundo dois aspectos: o aspecto temporal que define as formas de cooperação (ou dos tipos de interação) possíveis, e o espacial que corresponde à disposição geográfica dos participantes [Fritzke95a].

Formas de cooperação

As formas pelas quais os participantes interagem em um grupo são de dois tipos: síncrona e assíncrona. A interação síncrona requer a presença simultânea de todos os participantes numa sessão. Já a interação assíncrona ocorre num intervalo de tempo maior e não requer a presença simultânea dos participantes. Existem também aplicações que misturam ambos os tipos de interações, como no caso de sistemas de conferência assíncronos que também suportam interações síncronas com alguns tipos de mídias (como voz ou texto, p. ex.).

Disposição geográfica

Os participantes de um sistema de Trabalho Cooperativo podem estar geograficamente distribuídos de várias maneiras: todos em um mesmo local (*co-located*), com suas estações de trabalho interligadas por uma rede local, e possivelmente compartilhando uma projeção de tela; em salas diferentes mas próximas (*virtually co-located*), interagindo por uma rede local e fazendo frequentemente uso de comunicação em tempo real de mídias como áudio e vídeo; em salas distintas dentro de um mesmo edifício (*locally remote*), interagindo também por uma ou mais redes locais, interligadas por uma rede do tipo *backbone*; em lugares distantes (*remote*) suportado por serviços de telecomunicações.

2.3.2 Classificação Funcional dos Sistemas de Trabalho Cooperativo

Pode-se encontrar atualmente sistemas de Trabalho Cooperativo em diversas aplicações e ambientes. A seguir, são apresentadas os modelos básicos de sistemas cooperativos [Fritzke95a] [Cosquer96].

Sistemas de Mensagens

Baseiam-se na troca de mensagens de texto de forma assíncrona entre usuários. Os sistemas de mensagens serviram inicialmente para a interação entre usuários de uma mesma máquina, mas atualmente, com o suporte de redes de longa distância, permitem a troca de mensagens entre

usuários de máquinas distintas. Os sistemas de mensagens são padronizados pelo CCITT, a partir da série de padrões X400 (*Message Handling Systems* - MHS). O correio eletrônico e os sistemas BBS (*Bulletin Board Systems*), são exemplos tradicionais de sistemas de mensagens.

Sistemas de Conversa Multi-Usuários

Este é um dos tipos mais simples de ferramentas para a colaboração. Normalmente os usuários se convidam uns aos outros para se juntarem a uma sessão imediatamente. Uma vez que a conexão é estabelecida os usuários compartilham uma janela e podem trocar mensagens textuais. Um tipo básico de comunicação síncrona para dois usuários está disponível através do programa *Talk* de UNIX™. Esta abordagem de conversação alfanumérica não provê muito suporte para a colaboração. Isto é porque as pessoas freqüentemente precisam de desenhos ou diagramas para expressar ou esclarecer suas idéias.

Sistemas de Conferência por Computador

Uma conferência consiste tipicamente de um conjunto de mensagens (textual ou multimídia) trocadas de forma assíncrona por um conjunto de pessoas localizadas em ambientes remotamente distribuídos, normalmente em torno de algum assunto de interesse comum. Conversações são entendidas como conjuntos de mensagens com ligações de comentários (como p. ex, respostas ou comentários a uma mensagem prévia). O acesso a uma conferência, assim como a organização e o controle de conversações (aprovação de novas mensagens, remoção das obsoletas), são normalmente controladas por um ou mais usuários com direitos especiais (moderadores).

A atual disponibilidade de redes de alto desempenho, assim como o surgimento do Mbone, tem possibilitado a existência de sistemas de conferências tempo-real (*real-time conferencing*) com interações síncronas entre usuários em localizações distintas. Sistemas de conferência baseados em ambientes gráficos de estações de trabalho (*desktop conferencing systems*), como no caso dos sistemas que usam as aplicações desenvolvidas para o Mbone, permitem interações de usuários através do uso de mídias como voz, imagem, gráficos e desenhos a mão livre.

Sistemas de Tomada de Decisão

Estes sistemas visam aumentar a produtividade no processo de tomada de decisão. A forma de cooperação é síncrona e a localização pode ser do tipo reunião frente-a-frente ou em sítios diferentes. Mecanismos de controle de acesso e de votação de cada usuário são necessários nestes sistemas.

Sistemas de Co-autoria e Argumentação

Estes sistemas auxiliam a autoria em cooperação de documentos (de texto ou com recursos multimídia) e suportam a negociação e a argumentação envolvidas no trabalho cooperativo. Editores distribuídos são sistemas de co-autoria, onde o documento final resulta das contribuições individuais e de negociações entre os participantes. O tipo de cooperação é geralmente síncrona e a localização geográfica vai desde frente-a-frente até remotamente distribuída.

Agentes Inteligentes

Os sistemas com agentes inteligentes baseiam-se na existência de participantes não humanos, como no caso de jogos para múltiplos usuários em computador e agentes inteligentes para o monitoramento de sessões de trabalho cooperativo.

Além destes modelos de sistemas cooperativos, também podem ser mencionados alguns outros exemplos de áreas de aplicações onde o trabalho cooperativo pode trazer benefícios consideráveis para a produtividade do trabalho.

Sistemas de Coordenação de Trabalho Cooperativo

Os sistemas de coordenação são sistemas de suporte à integração e ao ajuste de ações dos indivíduos no sentido de se alcançar um objetivo comum. Em geral estes sistemas permitem a um usuário visualizar suas ações assim como ações relevantes de outros usuários, no contexto do objetivo comum, assim como gerar lembretes e alertas aos usuários.

Suporte à Engenharia de Software Concorrente

Este tipo de sistema tem como objetivo principal diminuir os custos devido às interações entre os membros de equipes de desenvolvimento de software. Estas aplicações podem beneficiar-se

de sistemas cooperativos do tipo síncrono e assíncrono, como sistemas de co-autoria e argumentação, sistemas de mensagens e sistemas de conferência.

Automação Industrial Integrada (CIM-*Computer Integrated Manufacturing*)

Sistemas de trabalho cooperativo vem também sendo utilizados com o objetivo de diminuir o intervalo de tempo entre a concepção inicial de um produto, até sua produção em larga escala. Neste caso, a interação entre engenheiros, especialistas, fornecedores e clientes, vem demandando ferramentas assíncronas, como correio eletrônico, e síncronas, como conferências em tempo real.

2.4 CARACTERÍSTICAS FUNCIONAIS DOS SISTEMAS DE TRABALHO COOPERATIVO

As características funcionais dos sistemas cooperativos estão relacionadas com aspectos sociais de grupos de trabalho, tendo influência direta no comportamento e eficiência tanto dos indivíduos, como do grupo como um todo [Ellis91][Fritzke95a]. As aplicações cooperativas apresentam, em conjunto com os aspectos temporais e espaciais anteriormente mencionados, os seguintes atributos:

Compartilhamento de Recursos e Coordenação

Nas aplicações de trabalho cooperativo com interações síncronas, deve existir um **compartilhamento de recursos**, dos atributos destes e das operações feitas sobre eles. Isto significa que os dados compartilhados devem estar sempre acessíveis aos usuários, assim como os usuários devem perceber imediatamente e de forma ordenada o efeito das operações de um membro do grupo sobre estes dados.

A **coordenação** visa minimizar as operações antagônicas nas interações entre os usuários e aumentar a produtividade do grupo. A complexidade da coordenação varia de acordo com o tamanho do grupo e a tarefa a ser desenvolvida. Normalmente, grupos pequenos requerem um grau menor de coordenação, devido a menor incidência de conflitos entre operações de usuários. Sessões de *brainstorming* e conferências são exemplos de tarefas que requerem mecanismos de coordenação extremos: enquanto no primeiro exemplo os participantes podem

interagir espontaneamente, no segundo caso apenas um participante se comunica enquanto os outros ouvem passivamente.

A característica interativa e coordenada dos sistemas cooperativos é uma importante diferença com relação às aplicações de sistemas distribuídos tradicionais, como sistemas de arquivos e bases de dados.

Grupos Dinâmicos

Os participantes entram em uma sessão de Trabalho Cooperativo segundo políticas preestabelecidas, através de convites, ou de requisições explícitas de entrada em um grupo. Ao entrar, um participante deve poder obter todas as informações públicas compartilhadas, para poder participar do trabalho nas mesmas condições que seus colegas. Em alguns casos é necessário o estabelecimento de níveis distintos entre os membros de um grupo com relação ao acesso às informações compartilhadas. A saída de membros ocorre espontaneamente, por expulsão ou por motivo de falha no sítio em que se encontram.

Mensagens Inter-relacionadas

As mensagens dos participantes podem estar inter-relacionadas, como é o caso de ligações de comentários em aplicações assíncronas do tipo conferência por computador. A inter-relação entre mensagens também pode apresentar-se na forma de restrições de ordenação de operações de usuários a nível do grupo. Como por exemplo, em aplicações síncronas do tipo edição distribuída, as operações de usuários sobre informações compartilhadas devem ser executadas na mesma ordem por todos os participantes, mantendo-se assim a consistência global do documento.

Interações Implícitas e Explícitas

Além de serem do ponto de vista temporal, síncronas e assíncronas, as interações entre usuários de uma aplicação cooperativa podem ser *implícitas* ou *explícitas*. O primeiro tipo trata da interação através das informações que compõem o objeto do trabalho conjunto, como no caso de um documento textual ou gráfico no processo de edição cooperativa. O segundo caso trata da comunicação direta entre os participantes de um grupo, através de canais de voz, vídeo (como em conferência tempo-real) ou através de gestos em reuniões “face-a-face”.

Interface Compartilhada

É desejável que as informações públicas compartilhadas sejam apresentadas de forma consistente aos usuários. O comprometimento com esta propriedade definida em [Stefik87] como WYSIWIS (*What You See Is What I See*), pode ser verificada em diferentes graus: desde o WYSIWIS estrito, onde todos os participantes visualizam informações idênticas, até as formas mais relaxadas, onde as informações sendo visualizadas por um participante não precisam ser as mesmas dos demais. Neste último caso pode-se ter, por exemplo, situações de autores lendo páginas diferentes de um mesmo documento, em uma sessão de edição cooperativa, ou ter-se diferentes vistas de um mesmo documento (como posições e formatos independentes dos ponteiros do mouse de cada usuário, diferentes contextos de cores, etc.).

2.5 SUPORTE PARA APLICAÇÕES DE TRABALHO COOPERATIVO

Entende-se como suporte para aplicações de trabalho cooperativo, o sistema computacional que oferece facilidades para a implementação de uma determinada classe de sistema cooperativo, ou até de um conjunto delas, segundo suas características funcionais.

A definição e a complexidade do suporte está fortemente ligada aos aspectos funcionais da aplicação em questão. Diferentes formas de cooperação e distribuições geográficas vão demandar diferentes tecnologias de coordenação e comunicação.

De forma geral, as características funcionais dos sistemas cooperativos anteriormente apresentadas demandam determinadas características do suporte, entre as que se encontram: *mecanismos de comunicação de grupo, mecanismos de ordenação de mensagens e suporte de tempo-real.*

As formas de coordenação das interações entre usuários, assim como a natureza dinâmica dos grupos compostos por estes usuários, evidenciam a necessidade de mecanismos de comunicação de grupo que permitam o uso de endereçamento de grupo, livrando o programador de detalhes de endereçamento dos processos individuais que compõem o grupo. A natureza dinâmica dos grupos que compõem sessões de trabalho cooperativo requer que o

gerenciamento de grupo seja também flexível no que diz respeito às entradas e saídas de participantes, e não estático com relação à configuração dos sítios de trabalho.

Os mecanismos de ordenação de mensagens têm que garantir a entrega das mensagens emitidas por membros de um grupo, respeitando relações de ordenação que ajudam a manter a consistência da sessão.

Os requisitos de tempo-real aumentam com o grau de sincronismo nas interações entre os usuários e com a utilização de mídias contínuas nessas interações. Em interações síncronas, requer-se que o sistema de Trabalho Cooperativo não torne o ritmo individual de cada membro mais lento do que se ele estivesse trabalhando de forma não integrada. Conseqüentemente, o sistema deve reagir de forma suficientemente rápida aos eventos globais disparados pelas atividades dos usuários. Por exemplo, ao ser modificado um texto ou um desenho em um editor cooperativo, cada usuário deve perceber essa modificação no contexto do documento global o mais rápido possível, para que cada um possa prosseguir com seu trabalho de criação, ciente de estar manipulando informações atuais.

A natureza de tempo-real das mídias contínuas (voz, vídeo, áudio, p. ex), reflete-se em requisitos temporais do suporte mais estritos que no caso de mídias discretas (textos, gráficos, etc.). As mídias contínuas são mais sensíveis a atrasos de transporte entre sua geração e apresentação em sítios remotos.

2.6 CONCLUSÕES

Neste capítulo foram apresentados inicialmente alguns conceitos básicos de *groupware*, seguidos de uma classificação dos sistemas de Trabalho Cooperativo, principalmente sob o enfoque dos aspectos temporais e espaciais destes sistemas. A seguir foram apresentadas características funcionais que, adicionadas aos aspectos temporal e espacial, definem atributos importantes relacionados com os sistemas de Trabalho Cooperativo. Ao final, são apresentados alguns requisitos gerais relacionados com o suporte computacional para aplicações cooperativas.

Este capítulo apresentou os aspectos teóricos básicos em relação ao Trabalho Cooperativo, com o objetivo de facilitar uma melhor compreensão, para o leitor, dos próximos capítulos.

CAPÍTULO 3

MBONE E IP MULTICAST

3.1 INTRODUÇÃO

A comunicação de grupo tem um importante papel dentro do trabalho cooperativo. Os mecanismos de comunicação de grupo permitem o uso de endereçamento de grupo, livrando o programador de detalhes de endereçamento dos processos individuais que compõem o grupo.

O conceito de endereçamento multicast para a comunicação de grupo na Internet foi proposto por Steve Deering em sua tese de doutorado na Universidade de Standford, e mais tarde desenvolvido na XeroxParc [Deering89]. Multicast para Internet foi adotado pela primeira vez no encontro da *Internet Engineering Task Force* (IETF) celebrado em março de 1992. Foi a partir do encontro da IETF em julho de 1992 que tomou o nome de “MBone” e com isto, a comunicação entre grupos, através da Internet, começou a se tornar realidade.

Inicialmente a rede MBone foi chamada de “rede virtual” devido ao fato que as capacidades de endereçamento multicast e roteamento multicast não existiam nos roteadores Internet que normalmente se produziam e nem nos sistemas operacionais da maioria das máquinas. Apesar da situação ter mudado de forma considerável, já que existem alguns roteadores multicast disponíveis no mercado e a maioria dos sistemas operacionais começaram a ter capacidades para suportar multicast, esses produtos ainda estão em minoria. O MBone, no entanto, está sendo transformado aos poucos de uma espinha dorsal virtual numa real, permanente e inteiramente disponível.

Neste capítulo começaremos apresentando alguns elementos básicos para compreender a base acima da qual se construiu a rede MBone. Depois disto apresentaremos a rede. Logo a seguir serão descritas as ferramentas multimídias sobre MBone e os testes feitos neste trabalho com as mesmas. Finalmente, serão discutidas as limitações das aplicações multimídias sobre a rede MBone, principalmente no que diz respeito ao uso em aplicações de Trabalho Cooperativo.

3.2 GENERALIDADES SOBRE REDES DE COMPUTADORES

3.2.1 Interconexão de Redes

Os computadores numa rede podem comunicar-se com computadores de outras redes através de roteadores. Como o nome sugere, os roteadores não só movem informação de um computador numa rede a outro computador em outra rede, mas também decidem que caminho a informação deve tomar para alcançar o *host* destino.

A comunicação e união entre várias redes diferentes constitui o que se conhece pelo termo de *Interconexão de Redes*.

O *Internet Protocol* (IP) é o primeiro protocolo de interconexão de redes usado para conectar computadores localizados em diferentes partes do mundo. É o protocolo usado pela Internet, que conecta redes de dados em mais de 50 países.

3.2.2 Mecanismo de Comutação de Pacotes Internet

A informação é transmitida na Internet na forma de datagramas ou pacotes. Estes pacotes são roteados desde a fonte até o destino, usando o mecanismo de comutação de pacotes. O pacote endereçado é roteado em direção ao destino sem que nenhum circuito virtual seja estabelecido entre a fonte original e o destino final. A rota que o pacote segue na rede não tem que ser a mesma sempre para o mesmo par de fonte e destino, esta rota é dinâmica [Tanenbaum94].

3.2.3 As Classes de Endereços Internet

Cada endereço Internet é um par consistindo do identificador de rede e do identificador de *host*. A parte do cabeçalho do pacote contém informação relacionada com o roteamento, tais como os endereços de rede dos *hosts* emissor e receptor.

Baseados nos identificadores de rede e de *host*, os endereços IP são classificados em 5 classes: A, B, C, D, e E. A figura 3.1 mostra o formato das 4 primeiras classes de endereçamento Internet. É o primeiro byte quem determina a classe do endereço Internet. Os endereços da classe D que são reservados para o Mbone, são chamados *endereços multicast ou de grupos*. Os endereços da classe E atualmente não estão definidos na Internet [Kumar95].

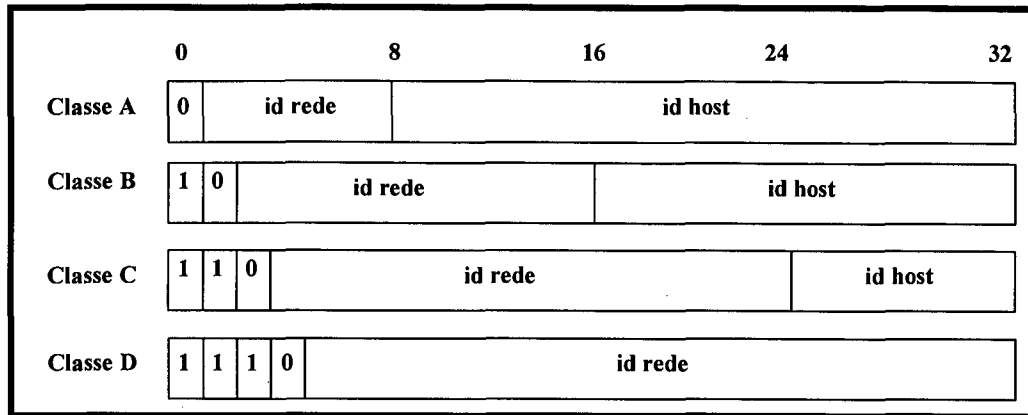


Fig. 3.1 As classes de endereços Internet

3.3 IP MULTICAST

O serviço *unicast* é o serviço clássico com o qual a maioria dos usuários de rede estão acostumados. É um tipo de transmissão ponto a ponto onde o pacote é transmitido do host fonte ao host destino. Esta é a forma padrão dos serviços que provêem protocolos de redes tais como o HDLC e o TCP [Tanenbaum94].

O serviço *multicast* é uma técnica usada para enviar cópias de um mesmo pacote a um subconjunto de todos os possíveis destinatários. Quando esse subconjunto é o conjunto de todos os possíveis destinatários a técnica é chamada de *broadcast*.

Com o serviço *broadcast* um pacote é transmitido a todos os hosts conectados à rede. Este serviço é freqüentemente indesejável, porque ele requer que as máquinas conectadas na rede executem algum tipo de processamento para determinar se estão ou não interessadas nos dados que estão sendo transmitidos. Além do mais, em redes grandes este tipo de serviço não é desejável porque o mesmo usa a largura de banda da rede em todo momento, independente se as sub-redes individuais estão interessadas na transmissão ou não.

O serviço *multicast* é o mais adequado para transmitir dados a um auditório potencialmente amplo na rede Internet. Somente aqueles *hosts* que estão interessados em um serviço *multicast* particular terão os dados roteados até eles. O conjunto destes *hosts* é chamado de grupo *multicast* e lhes é associado um endereço específico.

Um *host* que deseja receber tráfego multicast destinado a um grupo, deve juntar-se ao endereço multicast correspondente. O conceito de endereçamento multicast para grupos de comunicação foi definido em [Deering89].

O *serviço multicast* não é orientado a conexão. Um datagrama multicast é liberado aos membros do grupo destino com a mesma confiabilidade “best effort” que um datagrama IP *unicast* standard. Isto significa que não é garantido que um datagrama multicast alcance todos os membros do grupo ou chegue na mesma ordem em que foi transmitido em relação aos outros pacotes.

Hosts individuais são livres para juntar-se ou abandonar grupos multicast a qualquer momento. Não existem restrições na localização física ou no número de membros de um grupo multicast. Um host pode ser membro de mais de um grupo multicast a qualquer momento e não tem que pertencer a um grupo para enviar mensagens aos membros de tal grupo.

3.3.1 Endereçamento Multicast

Um endereço multicast IP é associado a um conjunto de receptores definindo um *grupo multicast*. O endereço multicast é usado como o endereço IP destino do pacote que vai ser transmitido a todos os membros do grupo [Deering89].

Um grupo multicast IP é identificado pelos endereços da classe D. Os endereços da classe D tem os 4 bits de maior ordem em “1110” seguidos pelos 28 bits que identificam o grupo multicast conforme mostra a figura 3.2

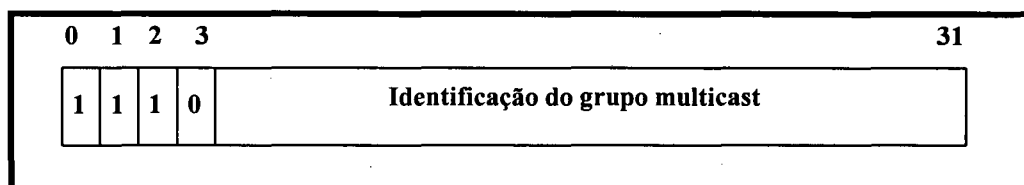


Fig. 3.2 Formato de endereço multicast classe D

A faixa válida para os endereços multicast IP expressos em notação ponto decimal vão desde 224.0.0.0 até 239.255.255.255. O endereço 224.0.0.0 é reservado e não pode ser usado por nenhum grupo multicast. O endereço 224.0.0.1 está reservado para o *grupo de todos os host* na

rede diretamente conectada. Os endereços a partir de 224.0.0.2 até 224.0.0.255, inclusive, são reservados para protocolos de roteamento e de baixo nível. O conjunto de endereços desde o 224.0.1.0 até o 239.255.255.255 estão abertos para uso geral.

Os endereços multicast não são endereços Internet físicos que estão amarrados a uma interface de rede física específica num determinado *host* físico. Eles são endereços de grupo lógicos que são dinâmicos por natureza, ou seja, eles existem enquanto existir um grupo de *hosts* Internet interessados em enviar e receber dados multicast.

3.3.2 Protocolo de Gerenciamento de Grupos Internet

Um *host* pode expressar seu interesse em ser membro de um determinado grupo multicast enviando uma mensagem *Internet Group Management Protocol* (IGMP) para juntar-se a tal grupo. O *host* pode então enviar e receber informação endereçada a esse grupo. O roteamento dos pacotes multicast é feito através do *Distance Vector Multicasting Routing Protocol* (DVMRP), o qual é um standard Internet [RFC 1075][Waitzman88]. Além do DVMRP outro esquema, chamado *Multicast Open Shorted Path Factor* (MOSF) [Moy94], é também usado no Mbone.

O *Internet Group Management Protocol* (IGMP) é usado para manter os roteadores multicast informados a respeito dos membros dos grupos presentes numa rede local particular. Um *host* IP numa rede local informa aos roteadores multicast vizinhos sobre os diferentes grupos multicast aos quais ele pertence.

O IGMP usa principalmente três tipos de mensagens [Deering89]:

- Tipo 1: Questionamento dos membros do grupo (*membership query*).
- Tipo 2: Relatórios dos membros do grupo (*host membership report*).
- Tipo 3: Atualizações de roteamento do protocolo DVMRP.

As mensagens do tipo "*membership query*" são geradas por um roteador *mrouter* numa sub-rede com capacidade multicast, e enviadas ao endereço de grupo 224.0.0.1 para determinar quais grupos de *host* têm membros nas redes diretamente conectadas a ele, como mostra a figura 3.3.

Quando um host recebe uma mensagem “query”, ele responde com uma mensagem do tipo “*host membership report*” para cada um dos grupos de hosts a que ele pertence. Para evitar muito fluxo de relatórios, cada host escolhe atrasos de tempo aleatórios para gerar os relatórios de cada um dos grupos a que ele pertence. Se durante esse período em que está aguardando para gerar o relatório, “escutar” outro relatório para o mesmo grupo, o host cancela o seu relatório para esse grupo. Caso contrário, o host transmite o relatório para o grupo fazendo com que todos os outros membros do grupo cancelem seus relatórios. Este procedimento garante que os relatórios estejam entregues num período de tempo limite e que somente um relatório seja gerado para cada grupo que tenha pelo menos um membro na sub-rede.

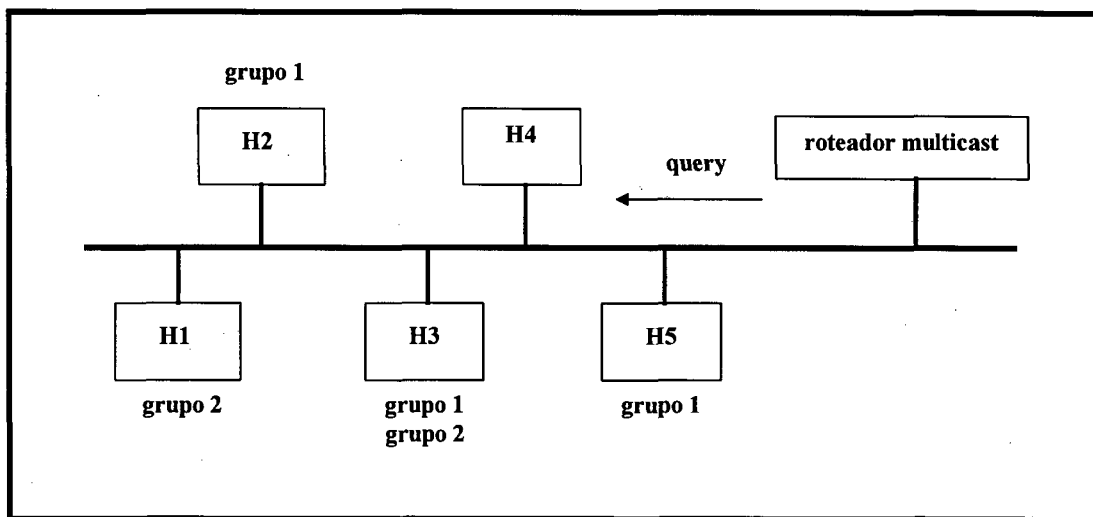


Fig. 3.3. Mensagem query no IGMP

Baseados na informação de quais são membros do grupo, o roteador é capaz de determinar qual tráfego multicast necessita ser retransmitido para cada uma das sub-redes. Os roteadores multicast usam esta informação junto com o protocolo de roteamento multicast para suportar IP multicast através da Internet.

3.3.3 Configurar o Sistema para Suportar Multicast

Antes de poder instalar e usar o MBone numa rede é necessário saber primeiro se os sistemas operacionais das máquinas que fazem parte da rede suportam *IP Multicast*. No caso dos mesmos não suportarem, as máquinas necessitam ser configuradas apropriadamente para suportar extensões para IP Multicast [Deering89].

3.4 A REDE MBONE

O MBone, abreviação de Internet Multicast Backbone, é uma rede virtual que usa o mesmo meio físico que a Internet. Consiste de um conjunto de roteadores Internet que compreendem os pacotes com formato de endereçamento Internet classe D e que sabem como roteá-los. Este esquema de endereçamento foi adotado pela primeira vez no encontro da *Internet Engineering Task Force* (IETF) em março de 1992 e foi chamado de MBone a partir de julho do mesmo ano [Casner92].

3.4.1 Roteamento Multicast

Os pacotes multicast são roteados de um ponto a outro do MBone usando um protocolo de roteamento multicast. Os roteadores multicast são conhecidos como *mroteadores* e tem a responsabilidade de distribuir e replicar o fluxo de dados multicast aos destinatários. A topologia MBone dos *mroteadores* é desenhada de maneira tal que facilite uma eficiente distribuição dos pacotes sem congestionar nenhum nó ou link de rede.

3.4.1.1 Protocolos de Roteamento Multicast

Um *protocolo de roteamento multicast* é um conjunto de padrões e parâmetros nos quais dois pontos de comunicação finais estão de acordo para o roteamento dos pacotes. Para rotear os pacotes de endereçamento de grupos multicast entre as diferentes sub-redes, diferentes esquemas de roteamento tem sido desenvolvidos no MBone. Estes protocolos de roteamento são padrões Internet e estão sendo desenvolvidos dentro dos roteadores.

Distance Vector Multicast Routing Protocol

Um dos primeiros mecanismos de roteamento, desenvolvidos para rotear pacotes de endereçamento de grupos multicast classe D, foi desenvolvido por Steve Deering na XeroxParc e conhecido com o nome de *Distance Vector Multicast Routing Protocol* (DVMRP) [Waitzman88]. Este protocolo de roteamento está implementado dentro do programa “mroued”, disponível como parte das extensões do kernel do sistema operacional para suportar IP multicast.

A especificação original do DVMRP [RFC-1075] foi derivada do *Routing Information Protocol* (RIP) e usava o algoritmo *Truncate Reverse Path Broadcasting* (TRPB). A diferença maior entre o RIP e o DVMRP está relacionada com o fato de que o RIP calcula a menor rota até um determinado destino, enquanto o DVMRP calcula a menor rota a partir de uma origem. A versão 3.7 do *mroteador* implementa o DVMRP, mas com o algoritmo *Reverse Path Multicasting* (RPM). Esta versão é um pouco diferente da original especificada no que diz respeito ao formato dos pacotes, formato dos túneis e tipos de pacotes, além de introduzir a poda de ramos inúteis introduzida em RPM.

Multicast Open Shortest Path First

MOSPF [RFC-1584] é uma extensão do protocolo de roteamento *Open Shortest Path First* (OSPF) [Moy94]. Uma rede de roteadores rodando o MOSPF pode transmitir pacotes multicast diretamente, enviando não mais de uma cópia sobre cada link, e sem necessidade do mecanismo de túneis¹.

OSPF é um protocolo unicast que requer que cada roteador construa uma base de dados da topologia de conectividade dos roteadores num domínio de roteamento. MOSPF estende o OSPF para fazer roteamento multiponto. MOSPF supõe que os roteadores adquiram informações sobre os grupos através do IGMP e mantenha uma base de dados local de grupos, contendo uma lista dos membros dos grupos locais. O conteúdo desta base de dados é divulgado para todos os demais roteadores do domínio (ou área) em anúncios do estado de enlaces e enviados por difusão. Isto permite que cada roteador mantenha informação completa sobre a localização de membros de todos os grupos.

Protocol Independent Multicasting

Tanto DVMRP como MOSP foram desenvolvidos para situações com densidade alta de membros de grupos. Quando os membros do grupo forem esparsamente distribuídos numa área geograficamente grande, os serviços multiponto destes protocolos não fazem uso eficiente dos recursos da rede. DVMRP periodicamente envia pacotes multipontos sobre muitos enlaces que não levam a membros do grupo, e MOSPF envia informação sobre a composição dos grupos por muitos enlaces que não levam a transmissores ou receptores. Adicionalmente, os dois

¹ O mecanismo de túneis é descrito no item 3.4.1.3

protocolos integram facilidades para a determinação das melhores rotas ponto-a-ponto, através de uma variação do RIP (DVMRP) ou OSPF (MOSFP).

Encontra-se em desenvolvimento pelo grupo de trabalho *Inter-Domain Multicast Routing* do IETF, o *Protocol Independent Multicasting* (PIM) [Deering90], um protocolo de roteamento multicast que utiliza informações sobre rotas ponto-a-ponto, obtidas através de um protocolo de roteamento unicast arbitrário. O PIM também distingue entre o roteamento para grupos densos ou esparsos (PIM-DM, PIM-SM). Um grupo denso é caracterizado por uma grande quantidade de estações e roteadores multipontos no domínio da comunicação, por exemplo, uma rede de campus, enquanto um grupo esparsos interliga nós *multicast*, através de caminhos onde nem todos os roteadores são multicast. O PIM provê mecanismos diferentes para cada um destes casos visando melhorar a eficiência principalmente quando os grupos são esparsos.

3.4.1.2 Time To Live (TTL)

Cada pacote multicast tem um valor de *Time to Live* (TTL) associado. O TTL facilita a distribuição do tráfego de dados multicast sobre o MBone. Este parâmetro permite ao usuário de aplicações especificar quão longe o tráfego multicast pode ir na Internet, ou seja, quantos roteadores IP são necessários pular para alcançar o destino. TTL atua como um mecanismo de limite para o roteamento de pacotes multicast [Kumar95].

Cada roteador multicast no MBone usa um valor limiar (*threshold*) associado para truncar o tráfego multicast. O limiar é definido como o mínimo TTL requerido para que um datagrama multicast seja repassado a uma interface dada ou túnel.

Cada pacote multicast tem um determinado TTL o qual será decrementado em 1 cada vez que passe de um roteador a outro no caminho para alcançar o destino. Para que um pacote multicast possa alcançar um roteador, o valor de TTL do pacote tem que ser maior que o valor do limiar do roteador.

Um valor de TTL de 16 limita o tráfego multicast ao âmbito de uma LAN. Os valores entre 127 e 255 permitem o envio do tráfego de dados multicast associado a toda a rede MBone. Valores de TTL entre 16 e 127 são também permitidos. Estes números sempre formaram parte do protocolo IP, mas não tinham sido usados antes da aparição do MBone.

É possível limitar a transmissão de eventos no MBone nos limites de uma LAN ou de um país dependendo do valor do TTL de maneira que o tráfego multicast não tenha que transitar por toda a parte da Internet que pertence ao MBone. Consideremos a situação mostrada na figura 3.4.

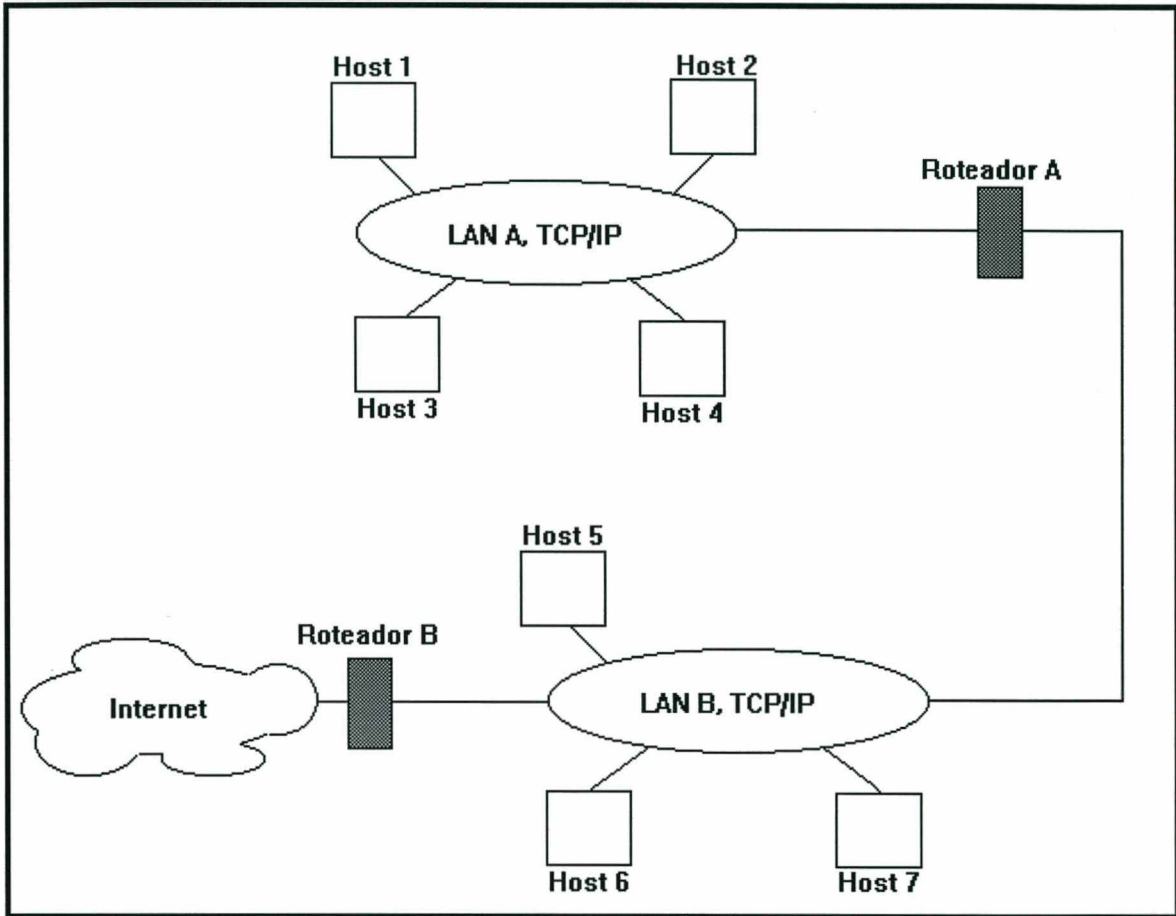


Fig. 3.4 Duas LANs rodando TCP/IP e interconectadas através do roteador A

Se o *Time to Live* (TTL) de uma sessão MBone for limitado a 16 ou menos, então o tráfego no MBone permanecerá dentro da mesma LAN (LAN A p.ex) onde ele foi criado. Se a sessão for criada, por exemplo, com um TTL de 17, isto fará com que o tráfego seja transmitido pelo roteador A aos host da LAN B, já que ele deixa passar os pacotes com TTL maior que o seu limiar (threshold). No entanto, esses pacotes não serão transmitidos pelo roteador B para o resto da Internet porque ao passar pelo roteador A este decreta o TTL associado ao pacote de um e o roteador B só deixa passar pacotes com TTL maior de 16.

3.4.1.3 Túneis

O MBone é operacionalmente suportado através do mecanismo de *tuneamento*, um esquema para a transmissão de pacotes multicast entre as sub-redes MBone. Isto é feito encapsulando os pacotes multicast dentro de pacotes IP regulares. Como resultado, os roteadores unicast IP que intervirem vêem os pacotes multicast como se fossem pacotes unicast e os retransmitem convenientemente [Kumar95]. Com o aumento de roteadores comerciais que suportem multicast, não vão ser mais necessários túneis para a transmissão dos pacotes multicast.

Túneis Truncados

Inicialmente os pacotes multicast IP do MBone eram roteados sobre o que se chamou *túneis truncados*.

No esquema de multicasting truncado, os pacotes multicast eram transmitidos a todos os roteadores multicast vizinhos sempre que o TTL fosse maior que o valor do limiar (threshold) do mrouter. Este esquema fazia com que mroteadores que não expressavam interesse em receber o tráfego multicast associado com um determinado grupo multicast o recebessem e em consequência o roteassem. O tráfego somente dependia do valor de TTL dos pacotes. Este não era um esquema muito eficiente, no entanto foi muito útil como mecanismo de teste na primeira etapa do MBone.

Túneis Pruned

Na atualidade a maioria dos túneis são “pruned”. *Pruning* em multicast é também conhecido como *multicasting verdadeiro* o qual significa que pacotes multicast não são retransmitidos a sítios ou mroteadores a não ser que eles tenham expressado interesse explícito em tais pacotes (via mensagens IGMP), endereçados a um grupo multicast específico e estejam dentro do âmbito especificado pelo TTL do emissor. O esquema TTL/threshold junto com o mecanismo de multicasting verdadeiro provêem um método eficiente de distribuir e permitir o acesso a mídia multicast sobre a Internet.

3.4.2 Tráfego no MBone

O tráfego no MBone é de vários tipos: Questão/Relatórios do IGMP, atualizações dos dados de roteamento do DVMRP do mroteador e dados das aplicações. Relacionaremos somente os

tráfegos vinculados com a aplicação porque o restante do tráfego gerado é mínimo (menos de 1 Kbps). Os dados da aplicação consistem de arquivos textos, gráficos e vídeo e áudio em tempo real [Macedonia94].

Áudio

A taxa proporcionada pelo hardware de áudio da maioria dos sistemas de computadores produzem 64Kbps de áudio PCM. A maioria das aplicações que estão disponíveis no MBone implementam compressão dos dados por software para diminuir a banda necessária a transmissão dos dados:

- 36 Kbps *Adaptative Pulse Code Modulation* (ADPCM)
- 17 Kbps *General Special Mobile* (GSM)
- 9 Kbps *Linear Predictive Coding* (LPC)

Vídeo

Para os vídeos com taxas de transmissão de frames menores de 30 frames por segundos que são os que prevalecem no MBone, a compressão, descompressão e apresentação dos frames de vídeo digital é feito por software. A taxa de dados que tipicamente se gera é de 125-128 Kbps. Taxas maiores podem ser usadas com um valor de TTL pequeno dentro de uma rede local.

Arquivos Textos e Gráficos

Os arquivos de texto e os gráficos são transmitidos no MBone com uma taxa de dados muito pequena. A largura de banda consumida por este tipo de dados está entre 5-6 Kbps.

Largura de Banda

Estima-se que a largura de banda total existente no MBone em qualquer tempo seja de 500 Kbps. Assim, podem existir várias sessões de áudio e vídeo transmitindo simultaneamente [Savetz96]. É considerado impraticável implementar o MBone sobre enlaces de capacidade inferior a 1,5 Mbps.

Protocolo de Reserva de Recursos

A limitada largura de banda para ser usada no MBone faz com que não possa ser garantida uma qualidade de serviço determinada durante o tempo total de duração de uma sessão, já que a qualidade do serviço vai depender do tráfego pela rede.

O *Resource Reservation Protocol* (RSVP) é a solução proposta para a reserva de recursos para estabelecimento de fluxos garantidos na Internet. Foi originalmente descrita em [Zhang93], e vem sendo aprimorado nas discussões do grupo de trabalho RSVP da IETF [Braden95].

RSVP trata do caso de entrega de dados entre um ou mais remetentes e múltiplos receptores para uma dada aplicação distribuída. A árvore (ou árvores) de distribuição é determinada por um algoritmo de roteamento multicast. É função dos roteadores multicast decidirem quais receptores devem receber os dados.

Um receptor interessado em receber os fluxos de um ou mais remetentes deve entrar para os grupos multicast apropriados. O receptor decide qual parte do fluxo ele gostaria de receber (por exemplo, imagem em preto e branco, ou colorida) e gera uma mensagem de *reserva*. A mensagem de reserva é propagada pela árvore de distribuição no sentido reverso na direção do remetente.

3.4.3 Topologia MBone

Dentro de um continente a topologia MBone será uma combinação de malha e estrela. As redes regionais serão unidas por uma malha de túneis que interconectam “mrollers”, os quais vão permitir agrupar as redes backbones que têm uma topologia de estrela com as redes regionais como mostra a figura 3.5 [Macedonia94] [Savetz96].

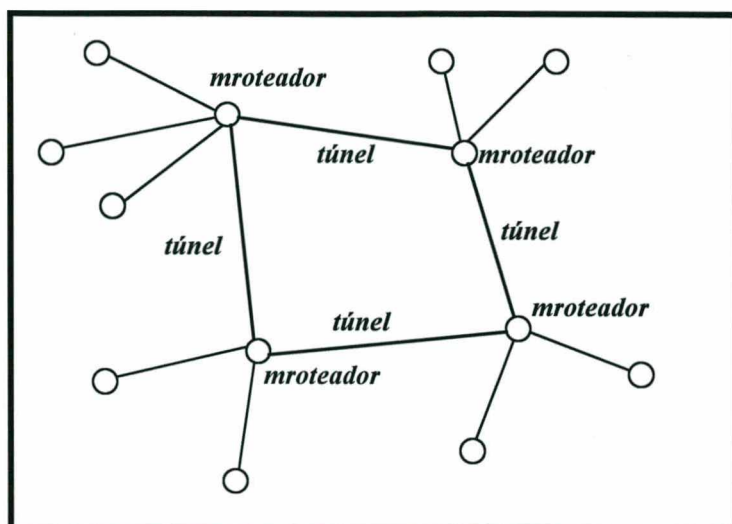


Fig. 3.5 Representação da topologia Mbone

A topologia MBone não é coordenada por ninguém em específico, pelo contrário, é um projeto cooperativo que combina o conhecimento dos participantes.

Foram estabelecidas listas de e-mail para coordenar a topologia MBone. A intenção é que quando uma nova rede deseja se unir, faça uma solicitação na apropriada lista MBone, então os participantes mais próximos respondem e cooperam para estabelecer os terminais do túnel apropriado.

3.5 FERRAMENTAS MULTIMÍDIAS SOBRE MBONE

Na atualidade existem ainda poucas classes de aplicações sobre Mbone. Elas são voltadas principalmente a videoconferências. *Visual Audio Tool* (VAT) e *Network Voice Terminal* (Nevot) permitem a um grupo de usuários fazer audioconferências, enquanto *NetVideo* (NV), VIC e IVS permitem fazer videoconferências. *Shared WhiteBoard* e *Shared Mosaic* são aplicações que estão disponíveis para conferências com transmissão de documentos. Todas estas aplicações usam IP multicasting para suportar o grande número de emissores e receptores ativos [Kumar95].

Podemos classificar as ferramentas existentes atualmente em quatro categorias: gerenciamento de sessões, vídeo, áudio e compartilhamento de documentos.

3.5.1 Gerenciamento de Sessões

Antes de fazer conferências de áudio, de vídeo ou com o WhiteBoard sobre o MBone, a sessão multicast necessita ser destinada, reservada e anunciada. Uma vez anunciada a sessão pelo criador do canal, cada máquina individual necessita ter acesso a tais anúncios e então ser capaz de incorporar-se ou sair de tais sessões.

Session Directory (SD)

A ferramenta SD é usada pelos usuários do MBone para reservar e destinar canais para a distribuição das mídias de um extremo e juntar esses canais aos diferentes usuários no outro extremo. O SD funciona como um *Guia de TV* do MBone [Casner94]. A janela do SD é mostrada na figura 3.6.

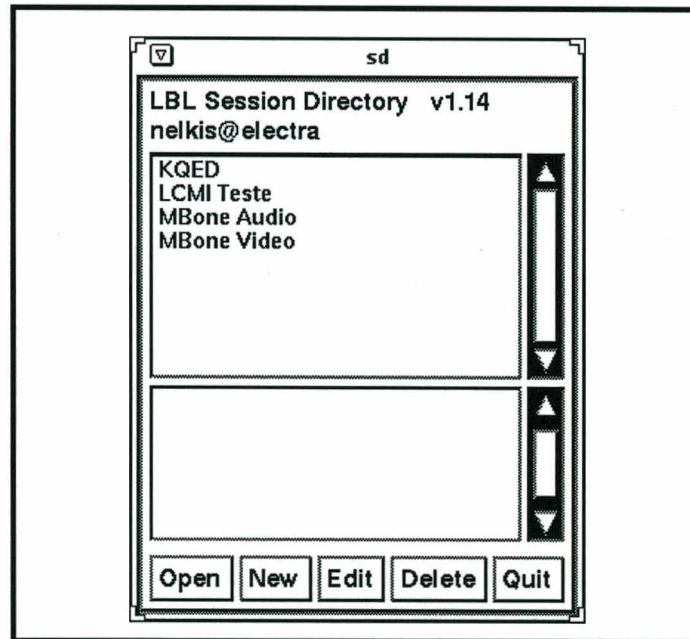


Fig. 3.6 Janela do SD

3.5.2 Vídeo

Entre as ferramentas de vídeo existem três que são as mais utilizadas: *INRIA Videoconferencing System (IVS)*, *NetVideo (NV)* e *VideoConference (VIC)*. Entre elas apenas a primeira engloba a transmissão de vídeo e áudio de forma integrada; as outras dependem de uma ferramenta auxiliar de áudio.

NetVideo

NetVideo (NV) é um dos softwares mais populares de acesso e entrega de vídeo em tempo real para o MBone. Este software permite o envio e recepção de imagens de vídeo através do MBone por múltiplos usuários. Ele funciona tanto ponto a ponto quanto multiponto. A interface do NV pode ser vista na figura 3.7. Todos os observadores e o emissor tem que estar conectados ao mesmo endereço multicast para poderem se comunicar através do modo multicast.

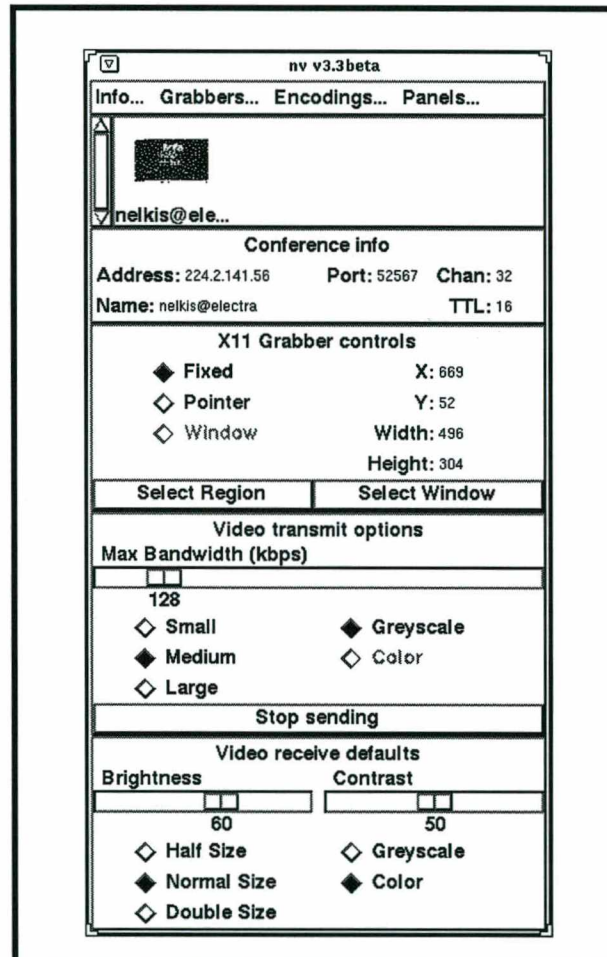


Fig. 3.7 Interface do NetVideo

NV usa um novo algoritmo de compressão de vídeo, o qual é desenhado especificamente para que o vídeo para redes consiga baixa vazão de dados e alta vazão de quadros. NV usa o protocolo *Real-Time Protocol, version 1* (RTPv1) como o protocolo de transporte de aplicação subjacente para transmitir e receber o fluxo de vídeo.

VideoConference

VideoConderence (VIC) foi desenhado com uma arquitetura flexível e extensível para suportar ambientes e configurações heterogêneas.

VIC baseia-se na versão 2 do *Real-Time Transport Protocol* (RTP). Este protocolo proporciona o suporte para a comunicação da mídia em tempo real. RTP é um protocolo a nível de aplicação e está implementado inteiramente dentro do VIC.

Para fazer uso das capacidades de conferências, o sistema deve suportar IP multicast e de preferência a rede deve estar conectada ao Mbone. VIC é compatível com RTPv1 e pode inter-operar com o NV e o IVS.

O VIC é bastante flexível em relação a possíveis codificações a serem usadas para a transmissão e identifica automaticamente qual delas foi utilizada no momento da recepção. Quando utilizada em conjunto com a ferramenta de áudio (VAT), esta pode automaticamente selecionar a imagem que está transmitindo áudio no momento.

INRIA Videoconferencing System

INRIA Videoconferencing System (IVS) é um software, que foi desenvolvido no INRIA, para transmitir áudio e vídeo usando uma workstation standard. IVS inclui codificação e decodificação PCM e ADPCM, assim como o codificador de vídeo H.261. Tanto a codificação como a decodificação do vídeo e do áudio são feitas por software. A figura 3.8 mostra os componentes da interface do usuário do IVS.

Esta ferramenta não tem implementado mecanismos que permitam o gerenciamento da videoconferência no que diz respeito à posse da palavra. O sítio que está nesse momento transmitindo os pacotes de voz é identificado com uma marca, os outros participantes têm que ficar alertas para quando ele deixar de transmitir, então eles transmitem. Nada impede a um sítio começar a transmitir independente de que outro já esteja transmitindo, o que traria como consequência perdas de informação.

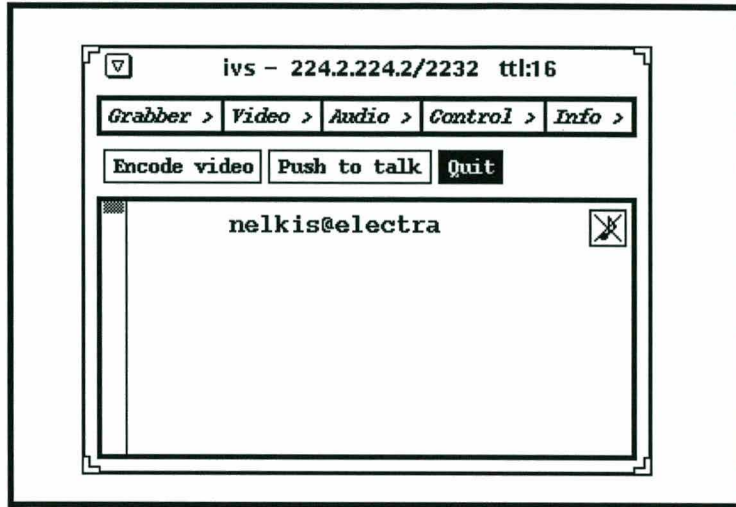


Fig. 3.8 Interface do IVS

3.5.3 Áudio

Visual Audio Tool

Visual Audio Tool (VAT) foi desenvolvido no Lawrence Berkeley Labs por Van Jacobson e Steve McCanne. VAT permite a dois ou mais host participarem em uma conferência de voz. A figura 3.9 mostra a interface.

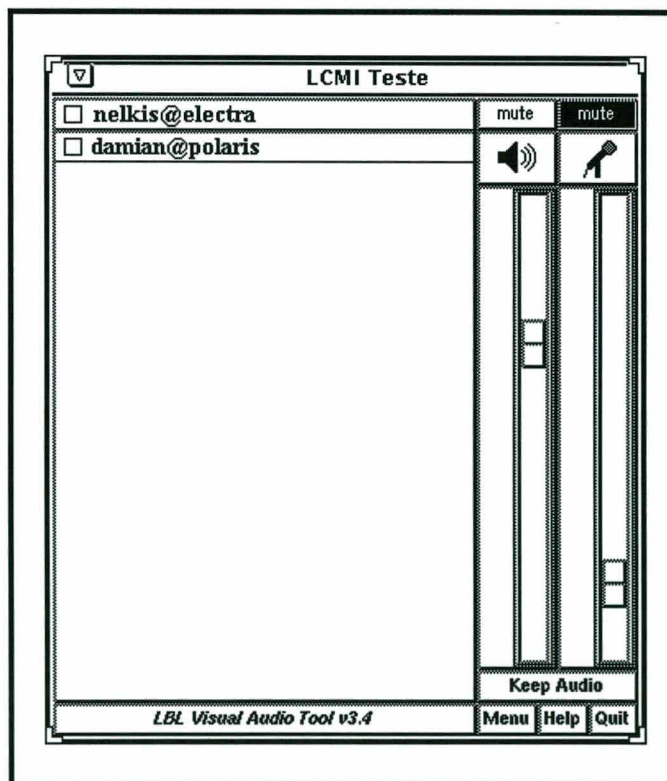


Fig. 3.9 Interface do Visual Audio Tool

O software somente requer dispositivos de entrada/saída de som, tais como microfone, um alto falante e uma placa de hardware de áudio.

VAT mostra a identidade de todos os participantes na sessão e destaca a identidade da pessoa que está falando. Os que falam e os que escutam têm um identificador consistente de um login e um nome de host Internet por default. Esta ferramenta não tem mecanismos para gerenciar a posse da palavra, por isso situações de concorrência provocam perdas de informação. No caso de uma aplicação cooperativa usando esta ferramenta isto poderia trazer como consequência o monopólio da palavra por parte de algum participante, perdendo-se a dinâmica da sessão.

3.5.4 Compartilhamento de Documentos

Tem-se a noção de que MBone somente trabalha sobre conexões com grande largura de banda e que o IP multicasting somente pode ser usado sobre redes de alta velocidade. No entanto, no MBone existem diversos protótipos que estão sendo usados e que trabalham com menos largura de banda que a requerida para as outras ferramentas. Um exemplo disto é o pacote de software WhiteBoard que será descrito a seguir.

WhiteBoard

WhiteBoard (WB) também foi desenvolvido nos laboratórios Lawrence Berkeley por Van Jacobson e Steve McCanne. WB é um bom exemplo de uma aplicação MBone que trabalha sobre a Internet em condições de largura de banda baixa.

WB permite a um ou mais sítios compartilhar um documento em tempo-real. O documento pode estar num formato Adobe PostScript ou em ASCII. Na figura 3.10 pode-se observar a interface da ferramenta.

Uma vez que o usuário decide carregar e compartilhar um documento, o documento é também transmitido aos outros participantes da sessão. O controle da sessão é fraco e informal, o qual significa que qualquer um dos participantes pode controlar a sessão compartilhada. Neste caso, controle significa que o usuário pode digitar texto, desenhar gráficos e fazer anotações. Para prevenir interferências no andamento de uma sessão, novas capacidades estão sendo

construídas no WB, tais como a capacidade de iniciar uma sessão em Modo Leitura, o qual evita interferência quando alguém se junta a uma sessão em andamento.

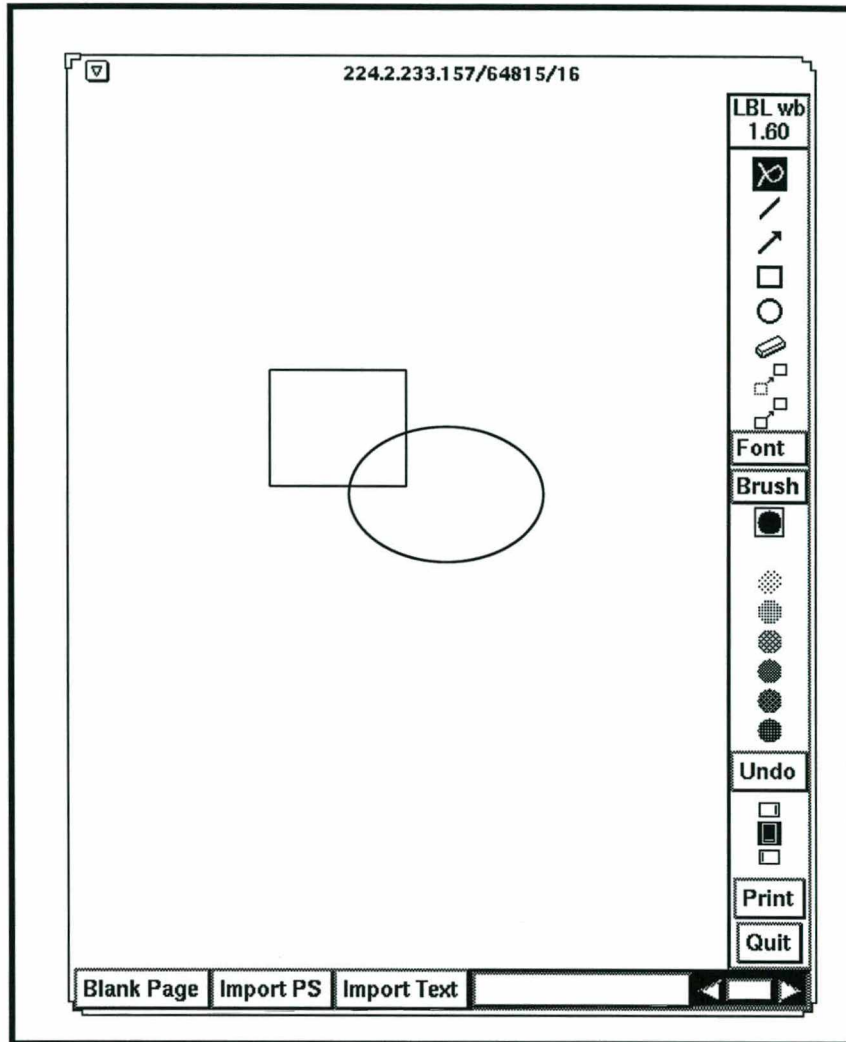


Fig. 3.10 Participantes compartilham uma sessão usando o WB

3.6 MONITORANDO O MBONE

Neste trabalho avaliou-se a utilização do Mbone para aplicações que exigem a transmissão de sinais de vídeo e áudio em tempo real sobre a Internet. Para tanto, usaram-se ferramentas disponíveis gratuitamente na Internet.

Foram monitoradas duas sessões Mbone correspondentes, com as ferramentas *mtrace* e *rtpqual*.

mtrace

O software *mtrace* indica os ramos da árvore multicast desde a fonte até o receptor para um grupo multicast particular, mostrando os diferentes *mroteadores* que tomam parte da transmissão dos pacotes desde a fonte até o destino.

rtpqual

O software *rtpqual* permite a hosts finais no MBone, monitorar a qualidade da recepção dos dados multimídias. O programa compreende as mensagens do protocolo *Real Time Protocol* (RTP) e as escuta, então computa e apresenta no monitor a seguinte saída:

T	Pkts	Loss	%	Late	Bytes	Pkts	Loss	%	Late	kB	Sender
---	------	------	---	------	-------	------	------	---	------	----	--------

A primeira coluna representa uma referência de tempo do sistema em segundos, de forma tal que eventos perdidos possam ser relacionados com outros eventos.

As colunas de 2-6 apresentam, respectivamente, a seguinte informação: pacotes de dados recebidos, pacotes de dados perdidos, porcentagem de perda, pacotes de controle atrasados, bytes recebidos (inclui pacotes de dados, de controle e atrasados).

As seguintes cinco colunas são totais acumulados para esse transmissor. As linhas são impressas somente quando existe chegada de pacotes.

Resultados do Monitoramento**Teste No. 1**

No dia 30 de agosto de 1996 foi transmitida pela rede MBone a defesa da dissertação de mestrado de José Eduardo M. S. Brandão (UF Paraíba).

A sessão durou aproximadamente duas horas. Foram usados os programas *Visual Audio Tool* (VAT) para a transmissão do áudio e *VldeoConferencing* (VIC) para a transmissão do vídeo.

A figura 3.11 mostra o caminho seguido pelos pacotes transmitidos, obtido pelo *mtrace*. O caminho mostra os diferentes mroters que tomaram parte na transmissão dos pacotes desde o

mroteador fonte *anjinho.dsc.ufpb.br* (150.165.2.21) até o destino *sol.lcmi.ufsc.br* (150.162.1.14) através dos túneis estabelecidos entre os diferentes pontos. O tempo total que demorou o software para achar o caminho foi de 1712ms.

```
Mtrace from 150.165.2.21 to 150.162.1.14 via group 224.2.239.143
Querying full reverse path...
 0 lcmi (150.162.1.14)
-1 lcmi (150.162.1.14) DVMRP thresh^ 1
-2 ceop4.rederio.br (146.164.10.4) DVMRP thresh^ 16
-3 gaia.coppe.ufrj.br (146.164.63.4) DVMRP thresh^ 8
-4 styx.iqm.unicamp.br (143.106.13.1) DVMRP thresh^ 32
-5 ska.cr-df.rnp.br (200.6.48.31) DVMRP thresh^ 64
-6 ? (150.165.2.249) DVMRP thresh^ 64
-7 anjinho.dsc.ufpb.br (150.165.2.21)
Round trip time 1712 ms
```

Fig. 3.11 Saída do *software mtrace* para o teste 1

A seguir analisa-se o comportamento de cada uma das mídias durante a transmissão.

Vídeo

A transmissão do vídeo se caracterizou pela grande quantidade de pacotes perdidos. A porcentagem de perda de pacotes teve uma média de 70%, chegando em alguns momentos a alcançar 96%. A quantidade de pacotes recebidos se manteve durante toda a sessão entre 1 e 2 pacotes a cada segundo. Estes valores fizeram com que a qualidade na recepção do vídeo fosse péssima, observando-se quase todo o tempo uma imagem fixa e, eventualmente, algum movimento muito lento.

Áudio

No caso do áudio, não existiram perdas de pacotes durante o tempo que se esteve monitorando a sessão. A quantidade de pacotes recebidos a cada segundo variou entre 15-26 pacotes, observando-se a cada 3 ou 4 minutos uma diminuição para uma taxa de 4-6 por segundo que, provavelmente, correspondem aos períodos de silêncio.

Apesar disto, a recepção do áudio pode ser qualificada de média, pois em alguns momentos a voz chegava truncada. Um mecanismo de sincronização intramídia poderia melhorar a apresentação desta mídia.

Teste No. 2

O dia 11 de setembro de 1996 foi monitorada a recepção de uma das videoconferências transmitidas dentro do calendário do evento *COMDEX INTERNET BRASIL'96*.

Os pacotes transmitidos seguiram o caminho entre os diferentes roteadores que fizeram parte da transmissão dos pacotes desde o host fonte *ofelia.lsi.usp (143.107.3.251)* até o destino *electra.lcmi.ufsc.br (150.162.14.9)* através dos túneis estabelecidos entre os diferentes pontos. O caminho foi completado em 712ms. A visualização deste caminho se fez através do *mtrace* conforme mostrado na figura 3.12.

```
Mtrace from 143.107.3.251 to 150.162.14.9 via group 224.2.226.167
Querying full reverse path...
 0 electra (150.162.14.9)
-1 atlas (150.162.14.1) DVMRP thresh^ 1
-2 ceop4.rederio.br (146.164.10.4) DVMRP thresh^ 16
-3 MBone.cerf.net (192.215.245.10) DVMRP thresh^ 64
-4 MBone.sdsc.edu (198.17.46.39) DVMRP thresh^ 64
-5 nccosc-MBone.dren.net (138.18.5.224) DVMRP thresh^ 48
-6 MBone.nsi.nasa.gov (192.203.230.241) DVMRP thresh^ 64
-7 dec3800-2-fddi-0.SanFrancisco.mci.net (204.70.158.61) DVMRP thresh^ 64
-8 dec3800-2-fddi-0.Denver.mci.net (204.70.152.61) DVMRP thresh^ 1
-9 dec3800-2-fddi-0.WestOrange.mci.net (204.70.64.61) DVMRP thresh^ 1
-10 dec3800-1-fddi-1.WestOrange.mci.net (204.70.64.45) DVMRP thresh^ 1
-11 lsi.poli.usp.br (143.107.103.5) PIM thresh^ 64
-12 ofelia.lsi.usp.br (143.107.3.251)
Round trip time 712 ms
```

Fig 3.12 Saída do software *mtrace* para o teste 2

A seguir analisaremos o comportamento de cada uma das mídias durante a transmissão.

Vídeo

A transmissão do vídeo se caracterizou por uma média de transmissão de 6.8 pacotes por segundo. Durante a transmissão se perderam poucos pacotes, com uma média de 0.4 pacotes perdidos por segundo.

O formato de codificação usado para a transmissão foi o NV com o qual se obteve taxa de transmissão citada de 6.7 frames por segundo. Entretanto, o uso do formato H.261 teria permitido alcançar taxas de transmissão de frames entre 2-4 vezes mais altas que as alcançadas com o NV.

Em geral, a qualidade na recepção do vídeo foi aceitável. A taxa de transmissão ficou ao redor de 13 KB/s. Uma taxa baixa para os valores de QoS desejados para uma aplicação com vídeo comprimido que deve ser de 2MB/s. No entanto, valores dessa magnitude são difíceis de serem obtidos numa rede como o MBone que apresenta um tráfego variável e imprevisível.

Áudio

No caso do áudio as perdas de pacotes durante o tempo em que se esteve monitorando a sessão foram maiores que no caso do vídeo, uma média de 1.3 pacotes por segundo. A quantidade de pacotes recebidos a cada segundo variou entre 9-28 pacotes, observando-se uma média de 23.4 pacotes por segundo.

A taxa de transmissão foi de 58 KB/s, uma taxa aceitável para o valor de QoS considerado para uma transmissão de voz com qualidade, que é de 64KB/s.

Pode-se comprovar que no sítio receptor o modo "Lecture Mode" resulta em uma melhor qualidade do áudio que no "Conference Mode". Isto deve-se a que no "Lecture Mode" a aplicação reserva um buffer de recepção maior, com o objetivo de reordenar os pacotes que chegam atrasados ou fora de ordem, isto traz como consequência uma demora maior entre o momento em que o pacote é enviado pelo sítio transmissor e o momento em que é apresentado no sítio receptor. Devido a essa demora, é mais conveniente usar o modo "Conference Mode" degradando a qualidade da apresentação do áudio e não perdendo a interatividade em sessões que requerem interação entre vários participantes.

No caso de aplicações de trabalho cooperativo este método de ordenação não seria adequado, já que este tipo de aplicação exige tempos de resposta baixos em relação às ações dos usuários, de forma que a modificação de um dado compartilhado por um determinado usuário seja percebida o antes possível pelo outros usuários participantes da sessão. Com um mecanismo

que leva em conta o tempo de validade das mensagens, este problema pode ser resolvido já que as mensagens seriam apresentadas tão rápido quanto fosse possível e nunca depois de seu tempo de validade expirar.

De forma geral, pode-se concluir que apesar de se receber em alguns instantes a voz truncada foi possível acompanhar a apresentação.

3.7 LIMITAÇÕES DAS APLICAÇÕES MULTIMÍDIAS COOPERATIVAS SOBRE MBONE

A seguir analisa-se algumas limitações observadas no uso com o MBone e nos testes efetuados, em relação à capacidade do MBone para o desenvolvimento de aplicações multimídias cooperativas.

Largura de banda

Atualmente existe uma largura de banda limitada para ser usada no Mbone: 500 Kbps a qualquer momento, o que leva a uma qualidade do serviço oferecido variável em dependência do tráfego pela rede no momento da transmissão. Neste sentido, novas tecnologias estão sendo criadas, como o *Resource Reservation Protocol* (RSVP) [Zhang93] que uma vez desenvolvido em toda a Internet levará a multimídia comercial sobre MBone a tornar-se uma realidade.

Ordenação das mensagens

Os testes feitos com as aplicações multimídias sobre MBone demonstraram a necessidade da existência de um suporte de ordenação que leve em conta que as mídias transmitidas são de tempo real.

Com a aplicação *Visual Audio Tool* (VAT) se comprovou que no modo de trabalho “Lecture Mode”, no qual a aplicação reserva um buffer de recepção maior para reordenar os pacotes que chegam atrasados ou fora de ordem, a qualidade na recepção do áudio aumentou notavelmente. No entanto, o uso deste modo traz como consequência uma demora maior entre o momento em que o pacote é enviado pelo sítio transmissor e o momento em que é apresentado no sítio receptor.

Como o método de ordenação usado pela aplicação neste modo não leva em conta os tempos de validade das mensagens, a demora na espera da chegada de pacotes atrasados traduz-se na perda de interatividade em sessões que requerem interação entre os participantes.

Controle de concorrência

Ao usar as aplicações multimídias, verifica-se a necessidade de colocar um mecanismo de controle de concorrência para o caso de sessões com fortes interações entre os participantes, porque conflitos não são tratados e isto conduz a perdas de informação e da dinâmica da sessão.

No caso das teleconferências isto não é tão grave, já que a maior parte do tempo é o conferencista quem está falando e o resto dos participantes se mantém na escuta e só em determinado momento acontecem sessões de perguntas/respostas.

No caso das aplicações multimídias cooperativas, isto pode levar a sessão cooperativa a um estado inconsistente, caso ocorram interações concorrentes ou conflitantes. Desta forma, faz-se necessário um mecanismo que garanta que operações deste tipo sejam percebidas por todos os participantes.

3.8 CONCLUSÕES

Neste capítulo foi apresentada a rede MBone e foram descritas as ferramentas multimídias que tem sido desenvolvidas sobre a mesma.

Em seguida foram apresentados os testes feitos neste trabalho com tais ferramentas. Estes permitiram concluir sobre as limitações das aplicações multimídias sobre a rede MBone, principalmente no que diz respeito ao uso em aplicações de Trabalho Cooperativo.

A partir dos aspectos discutidos neste capítulo verifica-se a necessidade de propor um suporte que garanta o cumprimento dos requisitos que as sessões de trabalho cooperativo exigem para poder desenvolver aplicações cooperativas multimídias na rede MBone, que será tratado no próximo capítulo.

CAPÍTULO 4

UMA ABORDAGEM PARA APLICAÇÕES COOPERATIVAS MULTIMÍDIAS SOBRE MBONE

4.1 INTRODUÇÃO

Conforme visto no capítulo anterior, as aplicações de trabalho cooperativo favorecem a integração dos usuários e permitem um aumento de rendimento na execução das tarefas, em relação ao trabalho individual isolado. O desenvolvimento de aplicações cooperativas na Internet oferece possibilidades de intercâmbio de idéias e de execução de tarefas em conjunto para um número maior de pessoas.

Apesar de terem sido desenvolvidas diversas aplicações¹ que permitem interações entre usuários através do uso de mídias como voz e imagem sobre MBone, as mesmas não possuem um suporte que garanta o cumprimento dos requisitos que as sessões de trabalho cooperativo exigem. O suporte para trabalho cooperativo deve possuir uma série de requisitos básicos, definidos a partir dos atributos que este tipo de aplicação apresenta.

Neste sentido, este capítulo vem propor uma plataforma de suporte que proverá importantes facilidades para o desenvolvimento de aplicações cooperativas síncronas que possam fazer uso de mídias contínuas.

Este capítulo apresentará o problema da ordenação nas aplicações de trabalho cooperativo. Em seguida serão colocados os limites do uso do MBone para aplicações de Trabalho Cooperativo, envolvendo dados multimídias. Uma solução baseada no protocolo proposto em [Baldoni96], para a ordenação de mensagens com limitações tempo-real em redes não confiáveis, é apresentada. Finalmente é descrita a plataforma proposta para aplicações cooperativas multimídias e a interligação entre os diferentes níveis da arquitetura desta.

¹ Uma descrição dessas aplicações pode ser encontrada no capítulo 3 deste trabalho.

4.2 O PROBLEMA DA ORDENAÇÃO NAS APLICAÇÕES DE TRABALHO COOPERATIVO

Em uma sessão de trabalho cooperativo os participantes interagem **compartilhando recursos** (como p.ex, texto, áudio, desenhos) e de forma **coordenada** [Fritzke95a].

Nas aplicações de trabalho cooperativo com interações síncronas, deve existir um **compartilhamento de recursos**, dos atributos destes e das operações feitas sobre eles. Isto significa que os dados compartilhados devem estar sempre acessíveis aos usuários, e que os usuários devem perceber imediatamente e de forma ordenada o efeito das operações de um membro do grupo sobre estes dados.

Por outro lado, a **coordenação** é indispensável para aumentar a produtividade do grupo, permitindo uma interação eficiente entre os usuários numa sessão de trabalho e visando minimizar as situações de conflito entre operações de usuários.

De forma geral, sistemas cooperativos demandam diversas características do suporte, entre as quais destaca-se a ordenação da entrega de mensagens emitidas por membros de um mesmo grupo. Cada sítio deve ter todas as mensagens na mesma ordem e ter as relações de causa e efeito mantidas, de forma que todos os sítios tenham a mesma visão consistente da apresentação. Este é um dos problemas fundamentais a enfrentar ao desenvolver aplicações cooperativas sobre a arquitetura Internet por não ter garantia de entrega de mensagens de forma ordenada.

4.3 LIMITAÇÕES DO MBONE PARA O DESENVOLVIMENTO DE APLICAÇÕES DE TRABALHO COOPERATIVO

Viu-se que o MBone, abreviação de Multicast Backbone, é uma rede virtual que usa o mesmo meio físico que a Internet, porém através de um sistema paralelo de roteadores e de endereços *multicast* providos pelo *IP Multicast*. A origem e as primeiras utilizações do MBone estão ligadas a transmissões dos encontros do *Internet Engineering Task Force* (IETF). A rede é

atualmente usada por pesquisadores para o desenvolvimento de protocolos e aplicações multipontos para a comunicação de grupos [Kumar95].

No Mbone, um datagrama multicast é liberado aos membros do grupo destino com a mesma confiabilidade *best effort* que um datagrama *IP unicast standard*. Isto significa que não há garantias que um datagrama multicast alcance todos os membros do grupo, além do que pode não chegar na mesma ordem em que foi transmitido em relação aos outros pacotes.

Torna-se necessário desenvolver mecanismos acima da tecnologia *IP Multicast* que permitam garantir a entrega da informação às aplicações de forma ordenada.

Diversas ferramentas têm sido desenvolvidas para a transmissão de áudio e vídeo sobre Internet e especialmente sobre MBone [Kumar95]. De forma geral, o uso destas ferramentas tem sido as videoconferências e a transmissão de seminários, onde a interatividade entre os usuários é fraca e nas quais as limitações que o MBone tem para o trabalho cooperativo podem ser toleradas.

No entanto, aplicações de trabalho cooperativo que permitam o tratamento de problemas complexos a partir da contribuição de cada um dos participantes da sessão de trabalho, requerem interações fortes entre os participantes. É preciso então solucionar as limitações que possui o MBone para as aplicações de Trabalho Cooperativo, as quais neste caso não podem simplesmente ser ignoradas.

4.3.1 Ordenação

Como é sabido, o meio de comunicação não permite a liberação das mensagens na mesma ordem em todos os sítios, porque as mensagens para diferentes sítios viajam através de diferentes caminhos. Por este motivo, o método usado tem que garantir que mensagens enviadas antes de outras sejam utilizadas nessa ordem pela aplicação, mesmo que cheguem fora de ordem no sítio receptor. Isto requer que determinada informação, que permita garantir a ordenação, seja incluída em cada mensagem. Como não existe relógio global nos sistemas distribuídos, a informação adicionada às mensagens deve indicar o conhecimento de que outras mensagens foram enviadas antes.

4.3.2 Tempo de validade das informações

Em algumas aplicações as informações a serem transmitidas são multimídias. Em conseqüência, o tempo influi na validade da informação; os fluxos de dados enviados por um dado usuário devem ser usados dentro de um determinado intervalo de tempo. Além disso, os dados multimídias que perdem seu *deadline* podem ser descartados. Isto representa uma degradação na qualidade do serviço fornecido, que deve ser minimizada.

O **tempo de validade de uma mensagem**, Δ , é definido como a duração do tempo físico, depois do tempo de envio, durante o qual os dados contidos na mensagem continuam a ter significado para o receptor. Uma mensagem que chegue ao receptor, depois deste tempo, não é útil e deve ser descartada pelo suporte. Nos sistemas multimídias, o tempo de validade de uma mensagem é a máxima demora que uma aplicação pode tolerar antes de degradar a qualidade dos seus serviços [Kopetz89].

4.4 AS SOLUÇÕES AO PROBLEMA DA ORDENAÇÃO COM TEMPO DE VALIDADE DAS MENSAGENS

4.4.1 Ordenação Causai

A noção de ordenação causal foi introduzida por Birman e Joseph [Birman87]. Esta noção se apoia na existência de um sistema de comunicação confiável (todas as mensagens serão certamente entregues) e não impõe limitações na entrega das mensagens (demoras de transmissão são imprevisíveis, mas finitas).

O *ordenamento causal* afirma que para qualquer processo a ordem na qual as mensagens são entregues não pode violar a relação *happened-before* do correspondente envio.

Definição

Relação *happened-before* (\rightarrow), é definida em [Lamport78].

A relação *happened-before* com notação " \rightarrow " é a ordem sobre um conjunto de eventos que satisfaz:

1. Se a e b são no mesmo processo (ou sítio) e a é produzido antes de b , então $a \rightarrow b$.
2. Se a é o evento de envio de uma mensagem por um processo e b é a recepção desta mensagem por outro processo, então $a \rightarrow b$.

Definição

Ordem Causal [Baldoni96]: Um sistema computacional distribuído \hat{E} respeita a ordem causal se para qualquer duas mensagens $m1$ e $m2 \in M(\hat{E})$ temos que:

se $m1$ foi enviada antes de $m2$ e tanto $m1$ como $m2$ tem o mesmo processo como destino, então $m1$ vai ser entregue antes de $m2$.

Na figura 4.1 mostra-se a relação de ordenação causal: $\{m1 \rightarrow m2 \rightarrow m4\}$

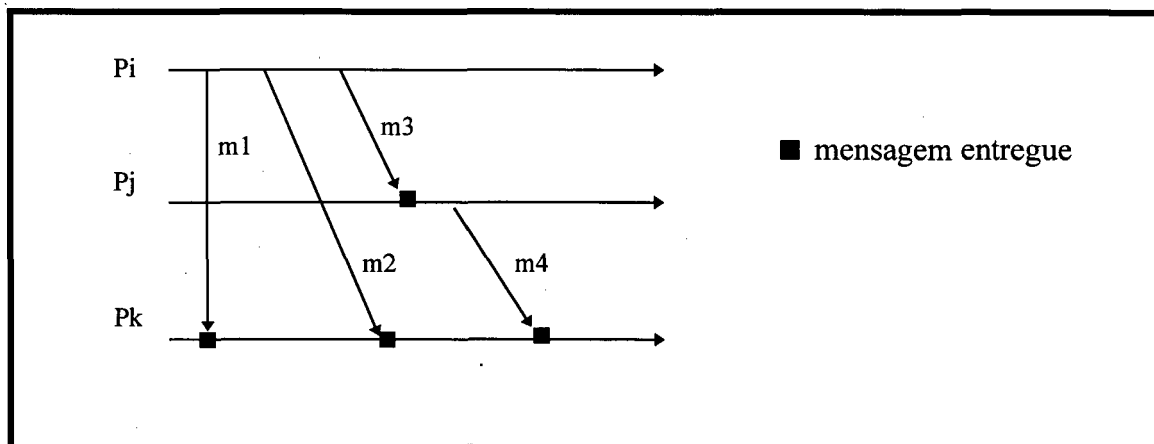


Fig. 4.1 Comunicação ordenada causalmente

Pela definição de ordenação causal pode existir um número arbitrário de mensagens envolvidas numa cadeia de dependências, já que não existe noção de tempo de validade destas.

Como vemos, a noção de ordenação causal não é adequada para a solução do problema apresentado porque ela pressupõe que as mensagens que são trocadas têm um tempo de validade ilimitado, o que não é o caso dos dados multimídias.

4.4.2 Ordenação Δ -Causal

4.4.2.1 Princípios e Definição

Como indicado previamente, a ordenação causal pode ser usada sempre em sistemas distribuídos confiáveis, para processos que trocam mensagens que têm tempo de validade ilimitado. Neste tipo de sistemas, cada mensagem possui associada uma condição de entrega que será verdadeira mais cedo ou mais tarde. Em sistemas onde as mensagens têm tempo de validade limitado deve ser levado em conta o seu eventual descarte se chegarem aos seus destinos depois de expirado seus *deadlines*.

Foram estes aspectos que motivaram Baldoni, Mostefaoui e Raynal [Baldoni96] a estender a noção de ordenação causal para o âmbito das redes de comunicação não confiáveis e com troca de mensagens com limitações tempo-real. Em particular, assumem que essas mensagens têm um tempo de validade limitado, Δ , depois do qual seus dados não podem ser usados pela aplicação e que algumas dessas mensagens podem se perder na rede de comunicação.

Consideramos que este protocolo é bem adaptado às aplicações cooperativas multimídias em redes Internet, já que estas redes tem características de serem não confiáveis. Além disso, as aplicações cooperativas que se pretende desenvolver acima da Internet vão incluir a transmissão de mídias como o áudio e o vídeo, que possuem um tempo de validade associado antes do qual a mídia deve ser apresentada pela aplicação em questão.

Definição

Δ -Causalidade: Um sistema computacional distribuído \hat{E} respeita a ordem de Δ -Causalidade se:

1. Todas as mensagens em $M(\hat{E})$ que chegam antes de expirado seus tempos de validade (ou seja dentro de Δ) são entregues durante seus tempos de validade; todas as outras mensagens não serão entregues (elas são descartadas ou foram perdidas no meio de comunicação).
2. Os eventos de entrega respeitam a ordenação causal.

A figura 4.2 mostra dois exemplos onde se emprega a noção de Δ -causalidade para ordenar as mensagens. Na figura 4.2.a as mensagens são entregues à aplicação no sítio P_k na ordem m_1 , seguida de m_3 . Neste caso se cumprem os itens 1 e 2 da definição de Δ -causalidade. As

mensagens que chegam dentro de seus *deadlines* ($m1$, $m2$ e $m3$), são entregues antes destes limites (item 1). As mensagens são entregues respeitando a ordem causal (item 2).

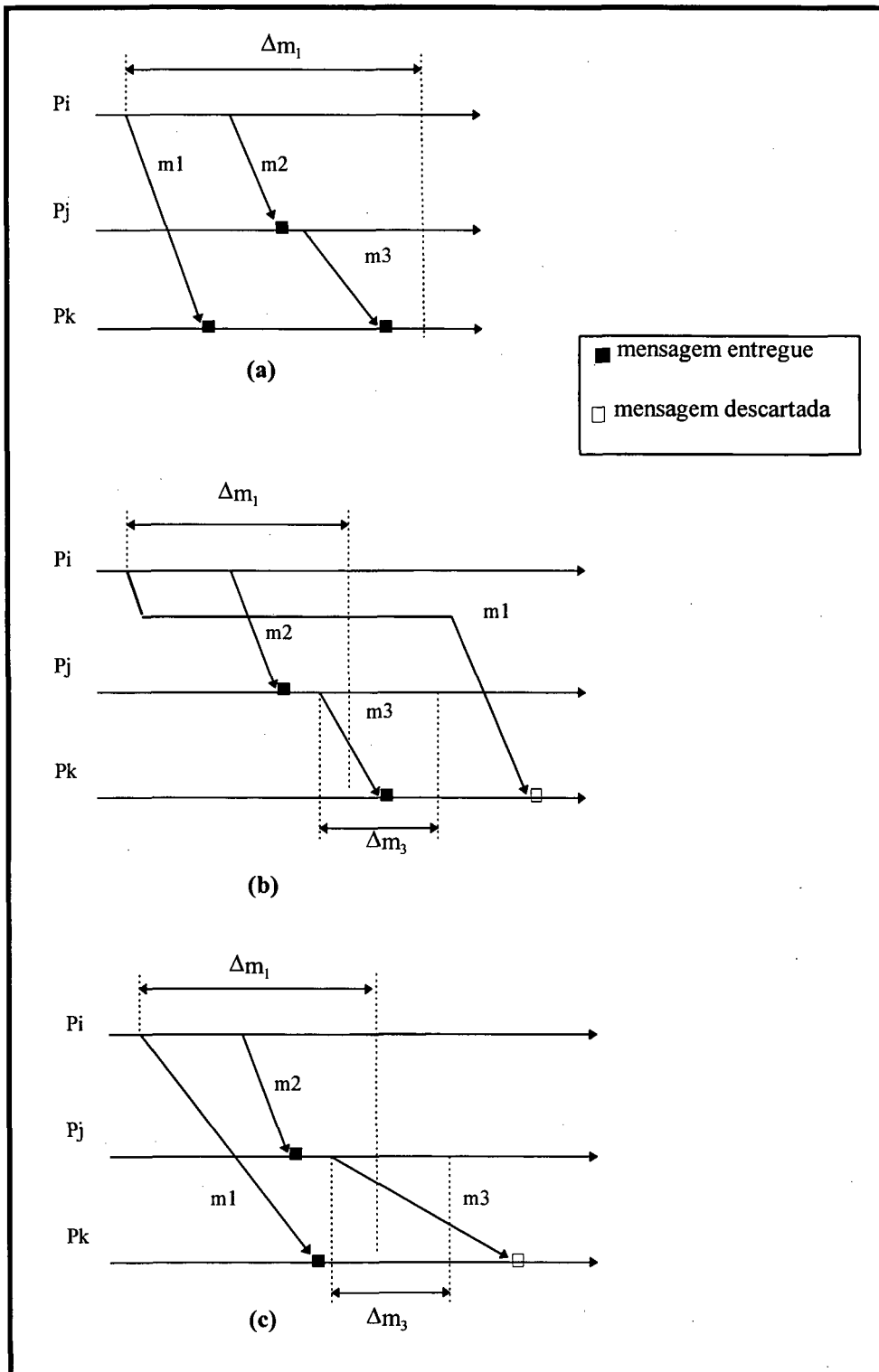


Fig 4.2 Comunicação com ordenação Δ -Causal

Na figura 4.2.b os eventos também são ordenados pela definição de ordem causal, já que $m1$ é descartada. Como se pode observar na figura, quando a mensagem $m3$ chega no sítio P_k a mesma é entregue dentro do seu *deadline*, e a mensagem $m1$, como chega depois do seu *deadline*, é descartada (item 1 da definição). As mensagens são entregues seguindo a ordem causal, já que $m1$ é descartada, só $m3$ é entregue à aplicação no sítio P_k (item 2 da aplicação).

A figura 4.2.c mostra uma situação em que a mensagem m_3 não é entregue à aplicação porque chega ao sítio P_k depois do seu tempo de validade.

4.4.2.2 O Uso da Δ -Causalidade em Aplicações Cooperativas Multimídias

Como tem sido expresso em [Wakeman93] em relação às redes de comunicações, um dos maiores desafios no desenvolvimento de aplicações multimídias em tempo-real é arcar com a perda de mensagens, a ordenação de mensagens e as demoras de transmissão arbitrárias. A ordem Δ -causal citada pela primeira vez em [Yavatkar92] é uma proposta para solucionar esses problemas.

A noção de Δ -causalidade é particularmente útil para aplicações de Trabalho Cooperativo, tais como teleconferências, onde os participantes necessitam trocar informação de vídeo e áudio em “tempo real” sobre uma rede de comunicação. Considera-se o fato de que as informações tempo-real têm um tempo de validade limite associado, depois do qual não podem ser usadas pela aplicação porque degradam a qualidade do serviço oferecido. Com o uso da Δ -causalidade para ordenar as mensagens, garante-se a entrega da maior quantidade de mensagens possíveis dentro do seu tempo de validade e descartam-se aquelas que chegam depois que expirou seu *deadline*.

4.4.2.3 A Necessidade do Tempo Físico para a Ordenação Δ -Causal

Para expressar as relações de causalidade entre mensagens sem limitações de tempo real, estampilhas ou matrizes de números de seqüência têm sido usualmente usadas nos protocolos de ordenação. Apesar dos relógios lógicos permitirem a garantia da ordenação causal das mensagens num nível lógico, eles só incrementam quando um evento ocorre (tais como na transmissão ou recepção de mensagens) e, portanto, eles têm pouca correlação com o tempo

físico e não podem ser usados para a Δ -Causalidade. Conseqüentemente, é necessário utilizar relógios físicos.

Apesar de ser impossível ter um relógio físico comum num sistema distribuído, particularmente em redes geograficamente distribuídas, existem protocolos de sincronização de relógios que permitem que todos os participantes num sistema distribuído estejam de acordo com um valor de relógio global virtual, cuja deriva (*drift*) em relação ao tempo físico permaneça sempre limitada.

4.5 ARQUITETURA PARA O TRABALHO COOPERATIVO EM REDES INTERNET

Para a comunicação entre membros de um grupo na Internet foi introduzido o conceito de endereçamento multicast e a rede MBone materializa hoje este conceito.

Entretanto, conforme visto no item 4.3, o MBone apresenta limitações para o desenvolvimento de aplicações do tipo Trabalho Cooperativo usando mídias contínuas. Essas limitações estão particularmente relacionadas com o fato de que o IP Multicast não garante ordenação das mensagens, e de que as aplicações de trabalho cooperativo exigem um suporte que garanta ordenação das mensagens, de maneira que todos os participantes tenham a mesma visão consistente da sessão de trabalho. Outrossim, deseja-se que as aplicações de trabalho cooperativo possam fazer uso de mídias como o áudio e o vídeo que possuem limitações tempo-real, e têm a noção de tempo de validade.

Neste sentido, propõe-se um protocolo de ordenação de mensagens construído acima do IP Multicast e que implemente a noção de Δ -causalidade já apresentada. Finalmente, é necessária a existência de um protocolo de sincronização de relógios, que proverá ao protocolo de ordenação de mensagens um tempo global virtual para expressar as relações de Δ -causalidade entre as mensagens. A figura 4.3 mostra os elementos da arquitetura proposta.

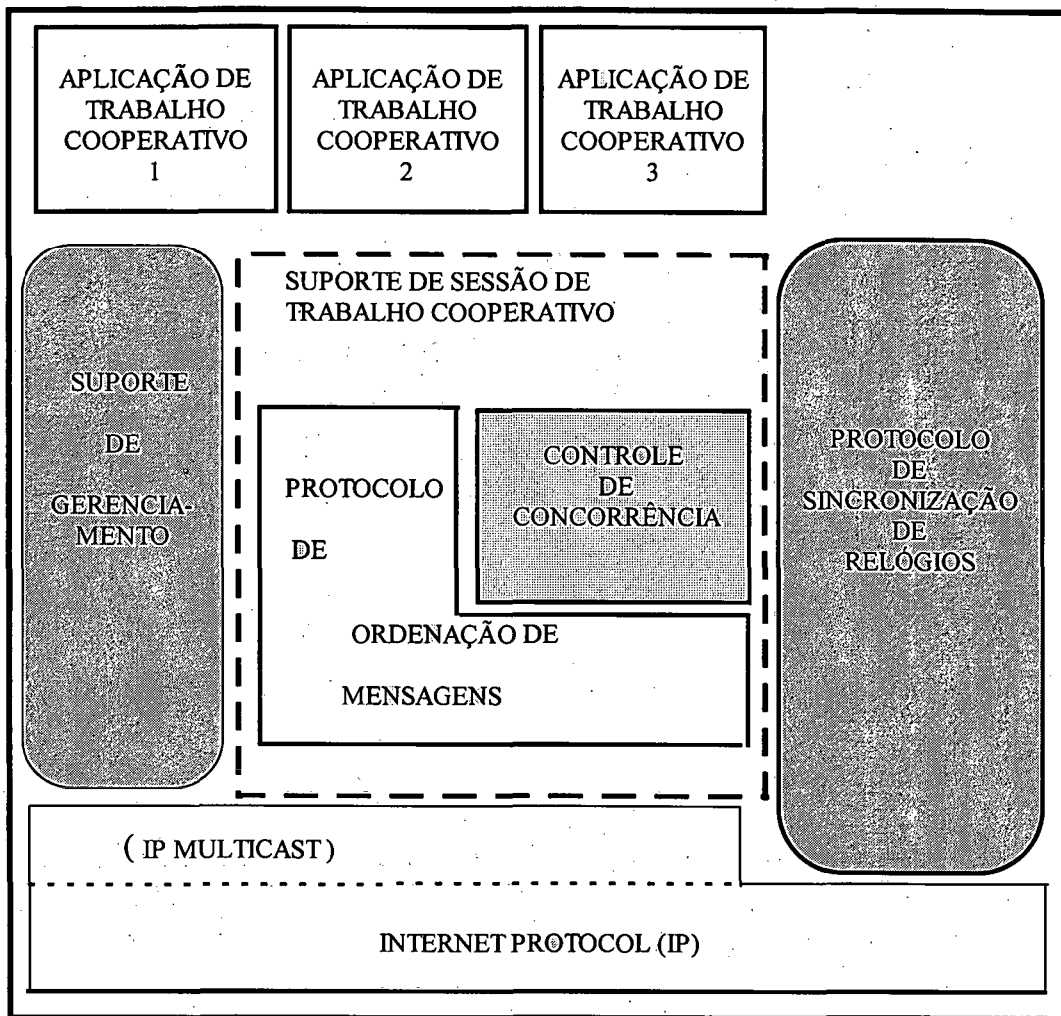


Fig. 4.3 Arquitetura para desenvolver aplicações de trabalho cooperativo sobre a Internet

O protocolo de ordenação de mensagens foi concebido para rodar sobre a Internet. Por este motivo ele faz uso dos elementos da *Application Programming Interface* (API) para IP Multicast, que têm sido introduzidos nos sistemas operacionais que suportam multicast para facilitar a programação de aplicações multipontos [Kumar95].

O protocolo possui um objeto encarregado da leitura do tempo do relógio local da máquina onde está rodando a aplicação. Este relógio é sincronizado com os outros relógios das máquinas que integram a sessão de Trabalho Cooperativo pelo protocolo de sincronização de relógios que será descrito a seguir.

Para enviar e receber a informação de maneira ordenada a aplicação deve fazer uso das classes de objeto que o suporte para a ordenação de mensagens lhe oferece.

Uma aplicação de Trabalho Cooperativo necessita ainda de um mecanismo de controle de concorrência para a coordenação das operações dos participantes e a interação entre estes numa sessão de trabalho. Dependendo da política de controle escolhida para a aplicação (mecanismo baseado em bastão, p.ex.) poderá ser necessário o uso do protocolo de ordenação de mensagens para a transmissão das mensagens de gerenciamento da sessão.

O suporte de gerenciamento oferece serviços que permitem a configuração dinâmica dos grupos *multicast*, assim como a manutenção dos mesmos.

A seguir, passaremos a descrever os elementos que integram o suporte de sessão de trabalho cooperativo que é o centro do nosso trabalho.

4.6 PROTOCOLO DE ORDENAÇÃO DE MENSAGENS

O protocolo descrito em [Baldoni96] que implementa a Δ -causalidade é apropriado para o nosso objetivo.

Devido as limitações de entrega impostas pela ordenação Δ -causal, quando uma mensagem chega, esta é descartada (se já passou seu *deadline*), ou pode ser atrasada até que a condição de entrega seja verdadeira. Uma mensagem que chega dentro de seu *deadline* tem que aguardar a chegada das mensagens que são causalmente precedentes, e que não tem ainda seus *deadlines* cumpridos.

A essência do protocolo consiste em definir uma condição de entrega (*Delivery Condition*) DC(m) associada com cada mensagem m de forma tal que m seja liberada tão rápido quanto possível, ou seja, quando todas as mensagens que precedem m já tenham sido liberadas, ou tenham sido perdidas, ou estavam ainda trafegando pela rede quando seus *deadline* expiraram.

Duas variantes do protocolo são apresentadas, uma variante para a comunicação ponto-a-ponto e uma variante multiponto para aplicações do tipo teleconferência ou *groupware*, onde sinais de áudio e vídeo de cada participante devem ser recebidos por todos os outros participantes.

Inicialmente será apresentada a variante geral (comunicação ponto-a-ponto) e logo a seguir são colocadas as modificações ao protocolo para a variante multiponto.

4.6.1 Descrição do Protocolo [Baldoni96]

- **Variáveis**

No protocolo [Baldoni96] é assumido que um mecanismo de sincronização de relógios entrega ao processo um valor único que representa o tempo global virtual. Cada processo vai ter uma variável *current_time* do tipo *time* que será continuamente atualizada pelo protocolo de sincronização de relógios subjacente. O domínio do tipo *time* é o conjunto dos números naturais.

Cada processo tem duas variáveis:

SENT_i : array [1..n, 1..n] of time

DEL_i : array [1..n] of time

A variável *SENT_i[x,y]* representa o conhecimento do processo *P_i* sobre o tempo de envio da última mensagem enviada por *P_x* a *P_y*. A variável *DEL_i[j]* representa o tempo de envio da última mensagem enviada por *P_j* a *P_i* e entregue por *P_i* à aplicação. Estas variáveis são iniciadas com um valor menor que o valor inicial da variável *current_time*.

- **Pseudo-Código**

Procedure Envio (m, i, j):

// *m* é a mensagem, *P_i* é o processo que envia, *P_j* é o processo receptor

Begin

send_time_m := *current_time*;

ST_m := *SENT_i*;

send (m, ST_m, send_time_m) to P_j;

$SENT_i [i,j] := send_time_m;$

End.

// Recepção e Ordenação das Mensagens

// As seguintes instruções são executadas com a chegada de uma mensagem no

// sítio P_i proveniente de P_j .

When $(m, ST_m, send_time_m)$ arrives at P_i from P_j :

Begin

if $send_time_m + \Delta < current_time$

then discard m

else wait $(\forall x : (ST_m[x,i] \leq DEL_i [x]) \text{ ou } (ST_m[x,i] < current_time - \Delta))$ //DC(m)

// As seguintes instruções são executadas atómicamente quando a condição $DC(m)$

// for verdadeira

$SENT_i [j,i] := send_time_m;$

$DEL_i [j] := send_time_m;$

$\forall x,y: SENT_i [x,y] := \max(SENT_i [x,y], ST_m [x,y]);$

deliver $(m);$

end if

End.

• Comentários sobre o Protocolo

Cada mensagem m enviada leva a seguinte informação:

ST_m é uma cópia da matriz $SENT$ do processo que está enviando a mensagem m , no tempo de envio da mensagem m .

$send_time_m$ é o tempo no qual a mensagem m está sendo enviada.

Além disso, temos que se ST_{m1} e ST_{m2} são as matrizes enviadas nas mensagens $m1$ e $m2$ respectivamente. Por definição: $ST_{m1} < ST_{m2} \Leftrightarrow \forall (x,y), ST_{m1} \leq ST_{m2}$ e $\exists (x,y), ST_{m1}[x,y] < ST_{m2}[x,y]$.

Quando uma mensagem m chega, esta é imediatamente descartada se já passou seu *deadline* ($send_time_m + \Delta < current_time$). Se m não é descartada, esta será entregue, seguindo a definição de ordem Δ -causalidade, dentro do seu *deadline*. A entrega é determinada pela *condição de entrega* $DC(m)$.

Uma mensagem m será entregue imediatamente assim que a condição $DC(m)$ seja verdadeira: todas as mensagens que precedem m já foram entregues (primeira parte da condição) ou seus *deadlines* já expiraram (segunda parte da condição). Formalmente a condição $DC(m)$ de entrega da mensagem m enviada para P_i pode ser expressa da seguinte forma:

$$\forall x : (ST_m[x,i] \leq DEL_i[x]) \text{ ou } (ST_m[x,i] < current_time - \Delta)$$

Finalmente quando a mensagem m é entregue, as variáveis DEL_i e $SENT_i$ são atualizadas de acordo com suas definições.

- **Variante Multiponto**

Nas aplicações do tipo teleconferências ou groupware está presente o conceito de grupo. Os participantes de uma sessão de Trabalho Cooperativo integram um grupo multiponto ao qual é associado um endereço *multicast*. Os sinais de áudio e vídeo transmitidos neste tipo de aplicações, devem ser recebidos por todos os outros participantes. As mensagens são transmitidas para o endereço do grupo e recebidas por todos os membros do grupo. Neste caso a estrutura de dados se simplifica, em cada um dos processos participantes, cada fila da matriz $SENT$ terá o mesmo valor, portanto em cada processo a matriz $SENT$ pode ser substituída pelo vector VT :

$$VT_i[x]: \text{array } [1...n] \text{ of time:}$$

onde a variável $VT_i[x]$ representa o conhecimento do processo P_i em relação ao tempo de envio da última mensagem enviada por P_x ao grupo de processos participantes da sessão.

Neste caso a condição $DC(m)$ pode ser expressa da seguinte forma:

$$\forall x : (VT_m[x] \leq DEL_i[x]) \text{ ou } (VT_m[x] < current_time - \Delta)$$

4.7 PROTOCOLO DE SINCRONIZAÇÃO DE RELÓGIOS

Como apresentado anteriormente, necessita-se para poder usar o protocolo de Δ -causalidade descrito anteriormente, que um protocolo de sincronização de relógios entregue aos processos um valor único que representa o tempo global virtual. Levando em conta que o objetivo é desenvolver aplicações de *groupware* sobre Internet, achou-se adequado o uso do *Network Time Protocol* (NTP) [Mills91], um protocolo de sincronização de relógios construído acima do *Internet Protocol* (IP), adequado para prover ao protocolo de ordenação de mensagens descrito anteriormente, um tempo global virtual.

4.7.1 Descrição do *Network Time Protocol* (NTP)

O *Network Time Protocol* (NTP) é usado para sincronizar o tempo de um cliente ou servidor, com outro servidor ou uma fonte de referência de tempo, tal como um modem receptor de rádio ou satélite. Este protocolo fornece o tempo ao cliente com uma precisão em torno do milissegundo em LANs, e até umas poucas dezenas de milissegundos em WANs, em relação ao servidor primário sincronizado com o *Universal Coordinated Time* (UTC) [Mills89].

O sistema distribuído de servidores de tempo NTP opera como uma subrede de servidores de tempo organizada hierarquicamente, os quais trocam periodicamente mensagens de atualização para ajustar o tempo do relógio local. Múltiplos servidores de tempos redundantes e diversos caminhos de transmissão são usados na subrede de sincronização, assim como algoritmos que escolhem as fontes e caminhos de sincronização mais confiáveis usando procedimentos de votação por maioria.

O sistema NTP consiste de uma rede de servidores primários e secundários de tempo, de clientes e de caminhos de transmissão interconectando-os. Um servidor primário de tempo é aquele que está diretamente sincronizado a uma fonte de referência primária, como por exemplo, um relógio atômico calibrado. Um servidor secundário de tempo se sincroniza a partir de servidores primários usando caminhos de rede, possivelmente compartilhados com outros serviços. Sob circunstâncias normais, a sincronização de relógios é determinada usando somente o servidor e os caminhos de sincronização mais precisos e confiáveis. Normalmente, assume-se uma configuração hierárquica, com as fontes de referência primárias na raiz e os

servidores diminuindo a sua precisão na medida que incrementam os níveis em direção às folhas.

Seguindo convenções estabelecidas pela indústria de telefonia, a precisão de cada servidor de tempo é definida por um número de *nível*. Aos servidores primários são associados o nível "1", e a cada nível subsequente em direção as folhas (servidores secundários) é associado um número maior que o precedente. O número adotado por cada servidor indica quão longe da fonte externa UTC o servidor está operando. A figura 4.4 mostra uma subrede NTP na qual os nós representam servidores de tempo com o número do nível associado determinado pela quantidade de saltos a partir da raiz. As linhas representam os caminhos de sincronização ativos e a direção em que flui a informação de tempo.

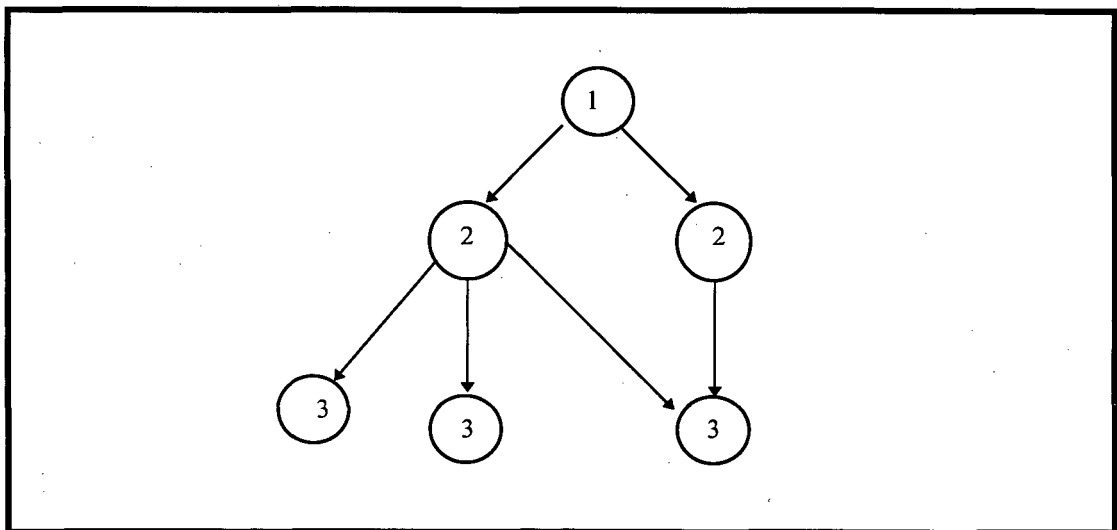


Figura 4.4 Subrede NTP

4.7.2 Funcionamento do NTP

NTP funciona acima do *Internet Protocol* (IP) e acima também do *User Datagram Protocol* (UDP). A abordagem usada pelo *NTP* para alcançar sincronização de tempo confiável desde um conjunto de servidores de tempo é diferente em relação a outros protocolos de sincronização. Em particular, *NTP* não tenta sincronizar os relógios uns com outros, ao invés disso cada servidor tenta se sincronizar ao *Universal Coordinated Time* (UTC) usando a melhor fonte disponível e o melhor caminho de sincronização até a fonte. Relógios sincronizados via *NTP* terão seus tempos bem próximos um dos outros, não porque os relógios

no grupo estejam sincronizados entre si, mas porque cada relógio do grupo está tentando sincronizar-se com o *UTC* através da melhor fonte a que ele tem acesso [Mills92].

O tempo é distribuído, como já foi colocado anteriormente, através de uma hierarquia de servidores *NTP*, onde cada servidor tem um número de nível associado, o qual indica quão longe da fonte externa *UTC* ele está operando. Os servidores do nível 1, são aqueles que estão no topo da árvore e que tem acesso a alguma fonte de tempo externa. Os servidores do nível 2 são aqueles que obtém o tempo de um servidor do nível 1, os servidores do nível 3 obtém o tempo a partir de um servidor do nível 2 e assim por diante. Para evitar longas cadeias de sincronização o número de níveis é limitado a 15.

Cada cliente na subrede de sincronização (o qual pode também ser um servidor para outros clientes de níveis superiores) escolhe exatamente um dos servidores disponíveis para se sincronizar, normalmente entre os servidores dos níveis inferiores aos quais tenha acesso. Esta configuração não é sempre ótima, já que *NTP* opera sob uma outra premissa: que o tempo de cada servidor deve ser visto com uma certa desconfiança. *NTP* na realidade prefere ter acesso a vários servidores, já que pode então aplicar um algoritmo de acordo para detectar diferenças por parte de qualquer um deles.

O modelo de serviço está baseado na entrega de tempo, a qual depende somente das medidas dos *offset* dos relógios, mas que não requer a entrega confiável das mensagens. O protocolo inclui condições para estimar o erro dos relógios locais em relação ao servidor de tempo ao qual este pode estar sincronizado.

NTP tem como propósito transportar informação relacionada com o tempo desde os servidores dos níveis inferiores até os servidores dos níveis superiores através da Internet, e além disso checar os relógios entre si para atenuar erros devido ao equipamento ou a falhas de propagação. A informação incluída numa mensagem *NTP* permite comparar o tempo do servidor com respeito ao tempo local e ajustar adequadamente o relógio local. Além disso, a mensagem inclui informação para calcular a precisão e confiabilidade, assim como para selecionar o melhor entre vários servidores.

4.7.3 Modos de operação

Existem vários modos nos quais os servidores NTP podem associar-se uns com os outros, de forma que o modo em que trabalha cada servidor do par indique o comportamento que o outro servidor pode esperar dele. A associação pode operar em um dos 5 modos seguintes: *simétrico ativo*, *simétrico passivo*, *cliente*, *servidor* e *broadcast*.

Modo simétrico ativo: Configurar uma associação neste modo indica ao servidor remoto que desejamos obter o tempo a partir dele e que também estamos dispostos a fornecer o tempo ao servidor remoto se ele precisar. Este modo é apropriado para configurações que envolvem um número redundante de servidores de tempo interconectados através de diversos caminhos de rede.

Modo simétrico passivo: Este tipo de associação é normalmente criada com a chegada de uma mensagem de um servidor remoto operando no modo simétrico ativo. Operando neste modo um servidor anuncia a sua disposição para sincronizar e ser sincronizado por outros servidores remotos.

Modo servidor: Este tipo de associação é criada depois da chegada de uma mensagem de um servidor operando no modo cliente. Operando neste modo o servidor anuncia a sua disposição para sincronizar servidores remotos. Neste modo normalmente opera o servidor de tempo de uma LAN.

Modo cliente: Configurar uma associação neste modo indica que se deseja obter o tempo a partir do servidor remoto, mas que não se está disposto a fornecer o tempo ao mesmo. As estações de trabalho numa LAN geralmente trabalham neste modo.

Modo broadcast: Um servidor operando neste modo envia mensagens periódicas independente do estado de alcançabilidade e do nível em que estão os outros servidores. Neste modo, geralmente trabalham os servidores de tempo das LANs, operando a altas velocidades num meio *broadcast*, anunciam a sua disposição para sincronizar servidores remotos mas não para serem sincronizados por algum deles.

4.7.4 O NTP Utilizado para Sincronizar os Relógios dos Participantes de uma Sessão de Trabalho Cooperativo

Devido ao número de nós existentes na rede Internet e à distância entre os mesmos, as soluções clássicas de sincronização de relógios físicos se tornam impraticáveis ou tem um desempenho ruim. Por este motivo tem surgido algumas propostas² de protocolos de sincronização de relógios que implementam algoritmos baseados em hierarquias para atenuar este problema [Veríssimo96] [Mills91].

A escolha do *NTP* para a plataforma proposta esteve determinada principalmente pelo fato do mesmo funcionar sobre o *Internet Protocol* (IP). Era necessário um protocolo de sincronização de relógios para sistemas distribuídos baseados em Internet, já que este é o âmbito de nossa aplicação.

Além do mais, a precisão do *NTP* em WANs está na ordem de umas poucas dezenas de milisegundos, portanto um pequeno limite superior ϵ (em torno dos 10-30 milisegundos de diferença entre os relógios de quaisquer dois participantes) pode ser assumido.

A maneira de levar em conta esse limite superior ϵ , no protocolo de ordenação de mensagens se assume de forma conservadora, um valor de $\Delta = D - \epsilon$. Onde temos que Δ é o tempo de validade da mensagem, D o *deadline* e ϵ a diferença de tempo máxima que pode existir entre os relógios dos participantes.

Para desenvolver uma aplicação cooperativa com a plataforma proposta é necessário que todos os participantes da sessão cooperativa tenham sincronizados seus relógios usando o protocolo *NTP*. Para isto, em cada uma das redes locais conectadas ao MBone onde exista interesse de participar em sessões de Trabalho Cooperativo, deve ser escolhida uma máquina para ser a servidora de tempo para a rede local. Esses servidores de cada subrede se associarão a servidores de tempo do mesmo nível, para garantir uma precisão de tempo bem próxima entre os participantes.

² Poucas propostas de protocolos de sincronização de relógios para redes de larga escala são encontradas na literatura.

4.8 MECANISMO DE CONTROLE DE CONCORRÊNCIA

As aplicações do tipo trabalho cooperativo precisam ter algum mecanismo de controle de concorrência para a coordenação dos participantes e permitir a interação entre estes numa sessão de trabalho e a interação entre estes numa sessão de trabalho. Existem diversas abordagens para a coordenação das interações dos participantes, que vão desde métodos sem nenhuma forma de controle de concorrência das atividades dos participantes até aquelas baseadas em controle de concorrência estrito.

Dependendo da política escolhida para o controle de concorrência será a relação deste módulo com o protocolo de ordenação de mensagens.

4.9 CONCLUSÕES

Neste capítulo foram apresentados os atributos que as aplicações de trabalho cooperativo exigem. Baseados nesses atributos foram colocados os problemas enfrentados na tentativa de desenvolver aplicações cooperativas multimídias sobre MBone.

Para a proposta de uma solução às limitações do MBone para o desenvolvimento deste tipo de aplicação tomou-se como base o protocolo proposto em [Baldoni96] e descrito neste capítulo. Finalmente foi apresentada a proposta de uma plataforma que permite a entrega ordenada de mensagens à aplicação cooperativa permitindo a criação deste tipo de aplicação sobre MBone.

No próximo capítulo será apresentado um exemplo de aplicação de trabalho cooperativo sobre MBone fazendo uso da plataforma proposta.

CAPÍTULO 5

UM EXEMPLO DE APLICAÇÃO DE TRABALHO COOPERATIVO MULTIMÍDIA SOBRE MBONE

5.1 INTRODUÇÃO

Como foi visto no capítulo 4, uma série de aspectos devem ser tratados antes de se poder desenvolver uma aplicação cooperativa multimídia sobre Internet, na qual possa haver uma interação forte entre os participantes e utilizar-se mídias contínuas.

Neste sentido, o capítulo anterior veio propor uma arquitetura de suporte de sessão de trabalho cooperativo utilizando um protocolo para a ordenação de mensagens segundo a noção de Δ -causalidade.

Este capítulo tem como objetivo apresentar um exemplo de aplicação de trabalho cooperativo sobre MBone fazendo uso da arquitetura proposta.

Inicialmente a arquitetura do suporte de sessão de trabalho cooperativo será apresentada. A seguir, serão detalhados os diferentes elementos que compõem a arquitetura e a forma com a qual estes se relacionam. Finalmente serão apresentados os testes do protocolo de ordenação implementado.

5.2 ARQUITETURA DA APLICAÇÃO COOPERATIVA MULTIMÍDIA

O cenário da aplicação escolhida consiste das ferramentas *Visual Audio Tool* (VAT) para transmitir áudio e o *WhiteBoard* (WB) junto com um *Telepointer* para a transmissão de texto, desenhos gráficos e anotações a mão livre entre os participantes da sessão.

A aplicação é composta das ferramentas citadas acima que foram embutidas em classes de objetos e fornecem, na sua utilização conjunta, as funcionalidades de uma edição cooperativa com auxílio de áudio.

O modelo de programação utilizado neste trabalho baseia-se no paradigma de orientação objeto. A figura 5.1 mostra a arquitetura da aplicação e do suporte de sessão de trabalho cooperativo construídos a partir deste modelo.

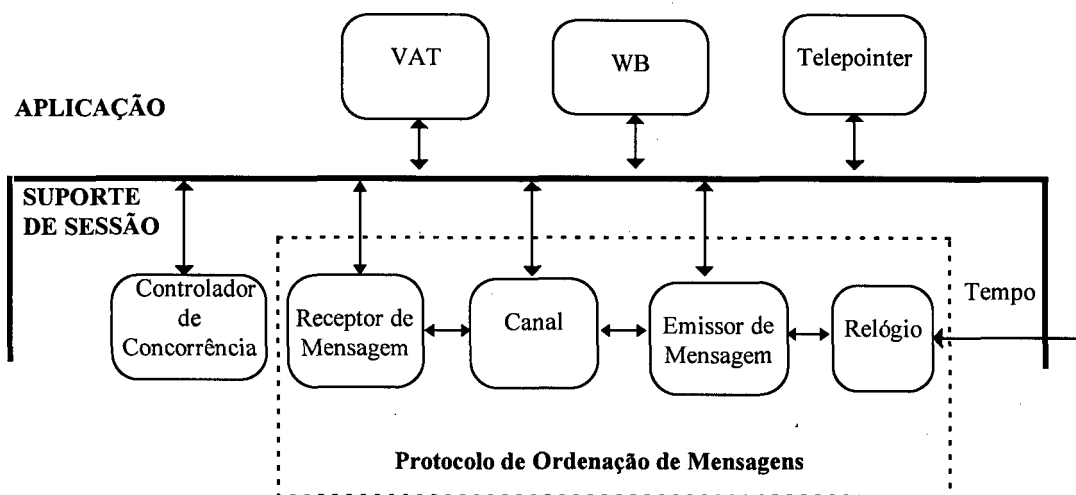


Fig. 5.1 Arquitetura da aplicação cooperativa multimídia em cada sítio participante

Como a ferramenta VAT é usada para a transmissão da voz que tem limitações de tempo para sua apresentação, a noção de Δ -causalidade se torna necessária. Conseqüentemente, esta ferramenta não vai mais utilizar seus próprios procedimentos de envio e recepção, mas fazer uso de objetos das classes *Emissor de Mensagem* e *Receptor de Mensagem* construídos a partir do protocolo de Δ -causalidade, para o envio e recepção das mensagens.

O WB é uma ferramenta cujo código não está aberto e que utiliza, para a comunicação, um protocolo de *multicast* confiável [Floyd95] embutido nela própria; portanto ela não precisa usar os serviços do protocolo de ordenação de mensagens. Além do mais, esta ferramenta não possui as facilidades de “telepointer”. Num trabalho futuro, pretende-se ter um *white board* com “telepointer” que permita a replicação consistente dos movimentos do ponteiro do

“*mouse*” para todos os usuários e possibilite apontar para partes do documento compartilhado, assim como deslocar objetos apresentados no *white board*.

Neste trabalho, como o código da ferramenta WB não está disponível, o “*telepointer*” desenvolvido terá como única funcionalidade a simples indicação de localização de objetos na janela do WB.

As mensagens que transmitem a informação do movimento do “*telepointer*” podem ter um tempo de validade associado e, nesse caso, é necessário a utilização dos serviços oferecidos pelo protocolo de ordenação de mensagens para a transmissão e recepção das mesmas.

Por outro lado, no que diz respeito ao Suporte de Sessão, a classe Controlador de Concorrência encapsula o mecanismo de controle de concorrência. Instanciações de objetos desta classe são feitas pelas aplicações cooperativas para a coordenação da sessão de trabalho cooperativo. Dependendo do mecanismo de controle de concorrência implementado, pode ser necessário usar os serviços do protocolo de ordenação de mensagens.

O Protocolo de Ordenação de Mensagens do Suporte de Sessão é o encarregado da ordenação das mensagens segundo a noção de Δ -causalidade. Este implementa o protocolo proposto em [Baldoni96] e descrito no capítulo anterior. A implementação do mesmo é descrita a seguir.

5.3 IMPLEMENTAÇÃO DO PROTOCOLO DE ORDENAÇÃO DE MENSAGENS

O Protocolo de Ordenação de Mensagens está composto por objetos das classes *Emissor de Mensagem*, *Receptor de Mensagem*, *Relógio* e *Canal*.

Este protocolo de ordenação de mensagens foi concebido para rodar acima do MBone, por este motivo ele faz uso dos elementos da *Application Programming Interface* (API) para *IP Multicast*, os quais tem sido introduzidos nos sistemas operacionais que suportam *multicast* [Kumar95].

Deste protocolo fazem parte uma série de elementos descritos a seguir.

5.3.1 Variáveis Globais

- **SENT[]** : cada sítio que roda uma aplicação cooperativa tem uma variável *SENT* que representa o conhecimento do processo sobre o tempo de envio das últimas mensagens transmitidas em *multicast* por cada um dos participantes da sessão. A dimensão da variável depende do número de participantes.
- **DEL[]**: esta variável tem armazenado os tempos de envio das últimas mensagens entregues à aplicação.

- **estr_mensagem**

```
{  
  short mensagem;  
  short emissor;  
  double tempo_envio;  
  double ST[ ];  
  char *dados;  
}
```

Esta estrutura de dados representa o conteúdo da mensagem que é transmitida e recebida pelo protocolo. Esta contém os dados da mensagem propriamente dita e o endereço da fonte de envio. O protocolo adiciona o tempo físico em que a mensagem está sendo enviada (variável *tempo_envio*) e uma cópia da variável *SENT* dentro da variável *ST*, informações estas usadas para garantir a ordenação, conforme apresentado no capítulo anterior.

Duas variáveis deste tipo são criadas: *pacote_envio* e *pacote_recepção*.

5.3.2 A classe Canal

Esta classe permite o estabelecimento do canal de comunicação com o grupo *multicast* da aplicação cooperativa através do seu método *Cria_sockets*.

O método *Cria_sockets*, cria os sockets de recepção e envio. Além disso, ele estabelece as opções correspondentes para a conexão dos *sockets* ao endereço de grupo para a comunicação *multicast*.

Aos *sockets* de recepção e envio são associadas as variáveis públicas *socket_recebe* e *socket_envio*, que são usadas pelos métodos das classes *Receptor de Mensagem* e *Emissor de Mensagem* para a recepção e envio dos pacotes *multicast*.

5.3.3 A classe Emissor de Mensagem

A classe *Emissor de Mensagem* possui o método *Envia_Mensagem* que será invocado pelas aplicações cooperativas e que implementa o procedimento *Envio* do protocolo de ordenação de mensagens descrito no capítulo anterior.

Quando este método é invocado, são passados os parâmetros que contêm informação referente aos dados multimídias e o endereço do processo que está enviando a mensagem. Este método inclui informações adicionais na mensagem para permitir que no sítio destino sejam ordenadas considerando a noção de Δ -causalidade. Essas informações são o tempo em que ela foi enviada e uma réplica da variável *SENT*. Esta última expressa o conhecimento do sítio emissor das últimas mensagens recebidas por ele e é usada pelo sítio receptor para saber quais mensagens foram enviadas ao grupo e que ele ainda não recebeu.

5.3.4 A classe Receptor de Mensagem

A classe *Receptor de Mensagem* possui os métodos *Ordena_Mensagem*, *Entrega_Mensagem*, *Checa_Fila* e *Atualiza_Fila*.

O método *Ordena_Mensagem* tem como funcionalidades: a leitura dentro de um ciclo infinito do canal de comunicação para a recepção das mensagens e o ordenamento conveniente das mesmas. Este método implementa as instruções correspondentes à recepção e ordenação das mensagens do protocolo de ordenação descrito no capítulo anterior. A instrução *deliver* deste protocolo corresponde ao envio da mensagem já ordenada à aplicação cooperativa, a qual é implementada pelo método *Entrega_Mensagem*.

O método *Ordena_Mensagem* lê a mensagem do *socket* de recepção e checa se o *deadline* da mensagem expirou, isto se faz comparando o tempo atual com a soma do tempo de envio da mensagem mais o *deadline*.

Se o *deadline* expirou, isto significa que o tempo de validade da mensagem acabou e que a sua apresentação não tem sentido porque degradaria mais a qualidade da apresentação, portanto a mensagem é descartada.

No caso do *deadline* da mensagem não ter expirado, fica-se aguardando que a condição de entrega seja verdadeira. A condição de entrega pode ser expressa como segue:

$$\text{if } (\forall x: (ST[x] < DEL[x]) \vee (ST[x] < \text{tempo_atual} - \text{Deadline})) \\ DC(m) = \text{verdadeira}$$

A primeira parte da condição checa se todas as mensagens que precedem *m* foram liberadas. Temos que, *x* representa o número de participantes da sessão, *ST[x]* é o tempo de envio da última mensagem enviada pelo sítio *x* ao grupo, segundo a visão do sítio emissor, e *DEL[x]* é o tempo de envio da última mensagem recebida do sítio *x* e entregue pelo sítio receptor à aplicação. Se *ST[x] > DEL[x]* significa que existe uma mensagem que foi enviada ao grupo e que ainda não foi liberada à aplicação pelo sítio receptor (pode ser que essa mensagem ainda não tenha chegado ao sítio receptor ou que tenha chegado, mas esteja aguardando por outras mensagens que a precedem para ser liberada).

A segunda parte da condição checa se as mensagens que precedem *m* já têm seus tempos de validade esgotados porque, em caso afirmativo, a mensagem será entregue à aplicação. Não tem sentido a mensagem ficar aguardando pelas outras porque quando essas chegarem seus *deadlines* terão expirados e serão descartadas.

No caso da condição de entrega não ser verdadeira, a mensagem é armazenada numa fila de espera onde aguarda que a condição *DC(m)* se torne verdadeira. O método *Checa_Fila* é o encarregado de checar quais mensagens da fila podem ser liberadas. Depois da liberação de uma mensagem, a fila de espera é atualizada usando o procedimento *Atualiza_Fila*.

A condição de entrega associada com cada mensagem m garante que m seja liberada tão rápido quanto for possível, ou seja, quando todas as mensagens que precedem m já tenham sido liberadas ou tenham sido perdidas ou estavam trafegando pela rede quando seus *deadlines* expiraram.

5.3.5 A classe Relógio

A classe *Relógio* constitui uma abstração do relógio local do sistema na forma de um objeto, cuja função consiste em oferecer o tempo do relógio local da máquina onde está rodando a aplicação cooperativa através do método *Tempo*, quando for requisitado pelo protocolo de ordenação de mensagens. A função deste método corresponde a da variável *current_time* no protocolo de ordenação. O tempo entregue por este método é um tempo global virtual já que o relógio da máquina local está sincronizado com os relógios das máquinas remotas usando o protocolo de sincronização de relógios NTP.

O protocolo de sincronização de relógios atualiza o tempo do relógio local de tempos em tempos a partir da informação do tempo obtida do servidor de tempo. *NTP* compara o tempo do servidor com relação ao tempo local e ajusta adequadamente o relógio local. O método *Tempo* lê o valor desse relógio. A figura 5.3 mostra a relação entre estes elementos.

O procedimento *Tempo* da classe *Relógio* é bem simples, ele só devolve no seu nome o valor que o relógio local tem nesse momento.

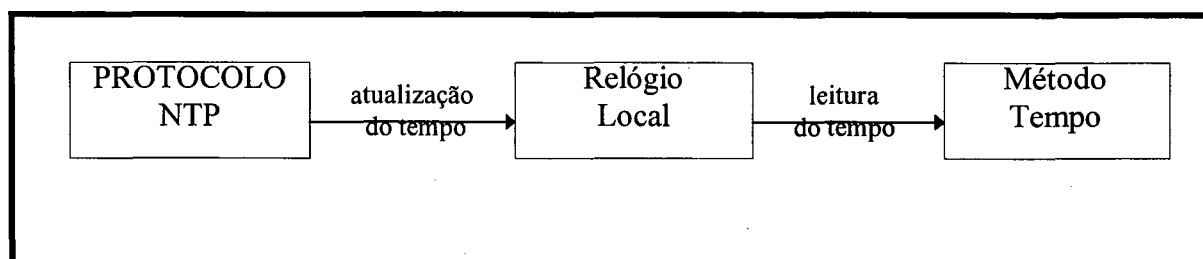


Fig 5.2 Relação entre a classe Relógio e o Protocolo de Sincronização de Relógios NTP

5.4 CONFIGURAÇÃO DO NETWORK TIME PROTOCOL (NTP)

Como se colocou anteriormente, para o funcionamento do suporte de ordenação de mensagens precisam-se ter os relógios dos participantes sincronizados. Para isto todos os nós MBone que vão participar das sessões de trabalho cooperativo, usando a plataforma que está sendo proposta, devem estar conectados à sub-rede *NTP*. Em cada uma das redes locais que estão conectadas ao MBone será escolhida uma máquina para ser a servidora de tempo para a rede local. Esses servidores de cada sub-rede se associariam a servidores de tempo público do mesmo nível.

5.5 EXPERIMENTOS SOBRE A APLICAÇÃO

5.5.1 Configuração do *NETWORK TIME PROTOCOL (NTP)* na rede local

A configuração do protocolo *NTP* na rede local foi feita da seguinte maneira:

O servidor de tempo da rede local foi associado a servidores de tempo primários (nível 1). Não é apropriado usar níveis mais baixos (2,3) porque a precisão diminui.

Foram escolhidos 3 servidores de tempo como fontes de sincronização, já que dessa forma *NTP* aplica um algoritmo de votação ao detectar diferenças entre eles.

A máquina selecionada para trabalhar como servidor de tempo para a rede local foi configurada no modo *servidor* do protocolo *NTP*. Dentro da rede local as máquinas sincronizadas ao servidor local operaram no modo *cliente*.

Com este procedimento conseguiu-se ter os relógios das máquinas sincronizados com o servidor remoto e em conseqüência ter seus tempos sincronizados, uns com outros. Em leituras feitas do tempo dos relógios das máquinas sincronizadas com o *NTP*, foi possível verificar que na configuração efetuada, existe um limite superior de 30ms de diferença entre os tempos dos relógios de duas máquinas quaisquer.

5.5.2 Teste do Funcionamento do Protocolo de Ordenação de Mensagens

Para testar o funcionamento do Protocolo de Ordenação de Mensagens criou-se uma aplicação simples que cria um grupo *multicast* e incorpora todas as máquinas que rodam a aplicação nesse grupo.

A aplicação usa um procedimento que gera números aleatórios para estabelecer, de forma aleatória em cada sítio, a frequência de envio das mensagens. O envio das mensagens equivale ao envio da informação do movimento do *Telepointer* na tela da máquina que roda a aplicação.

As mensagens recebidas são ordenadas conforme a noção de Δ -causalidade e apresentadas na tela de maneira ordenada, representando a forma em que as mesmas seriam entregues a uma aplicação multimídia encarregada de apresentar pacotes de vídeo ou áudio.

A aplicação faz uso de um procedimento para gerar demoras aleatórias virtuais, com o objetivo de simular atrasos na rede e comprovar que as mensagens que chegam quando expirado seu tempo de validade são descartadas.

Verificou-se o descarte das mensagens com tempo superior ao tempo de validade. Para as mensagens não descartadas, comprovou-se o mesmo ordenamento em todos os sítios, o que significa que estes terão a mesma visão consistente da sessão de trabalho cooperativo.

Para que a aplicação cooperativa multimídia proposta seja completada é preciso mudar o código fonte da ferramenta VAT de maneira que esta não faça mais uso de seus próprios procedimentos de envio e recepção, mas faça uso de objetos das classes *Emissor de Mensagem* e *Receptor de Mensagem* para o envio e recepção de mensagens. Esta mudança não será objeto deste trabalho por razões das limitações de tempo deste.

5.6 CONCLUSÕES

Este capítulo apresentou o cenário de uma aplicação cooperativa multimídia. Em seguida foi apresentada a arquitetura desta aplicação e foram descritos os diferentes elementos que integram o Protocolo de Ordenação de Mensagens.

Desta forma, como resultado final foi implementado um protocolo de ordenação de mensagens, garantindo às aplicações cooperativas que os participantes da sessão tenham uma visão consistente dos dados sendo compartilhados e descartando as mensagens que estão fora do seu tempo de validade.

Comprovou-se ainda que o protocolo de sincronização de relógios *NTP* tem uma precisão adequada para aplicações que envolvem tempos de validade da ordem de grandeza das mídias de áudio e vídeo.

CAPÍTULO 6

CONCLUSÕES E PERSPECTIVAS

O desenvolvimento de aplicações de Trabalho Cooperativo vem sendo, atualmente, amplamente discutido na literatura, devido à importante meta à qual estes sistemas se propõem que é a de permitir que grupos de indivíduos possam colaborar e obter resultados melhores e de maneira mais eficiente que aqueles obtidos pelo esforço individual. Entretanto, a tarefa de construir aplicações deste tipo demanda suportes que permitam ao programador preocupar-se essencialmente com a semântica da aplicação, e não com os mecanismos empregados para obter os resultados pretendidos.

Para poder desenvolver aplicações cooperativas sobre a Internet foi necessário, em primeiro lugar, ter mecanismos de comunicação de grupo que permitissem o uso de endereçamento de grupo, livrando o programador de detalhes de endereçamento dos processos individuais que compõem o grupo. Neste sentido foi instalada uma sub-rede MBone no laboratório de maneira a dotar às máquinas com as facilidades do endereçamento multicast para a comunicação de grupo na Internet.

Os testes feitos com as ferramentas multimídias sobre MBone demonstraram que apesar destas permitirem interações entre usuários através do uso de mídias como voz e imagem, as mesmas não possuem um suporte que garanta o cumprimento dos requisitos que as sessões de trabalho cooperativo interativas exigem.

Este trabalho apresenta um esforço no sentido de projetar e implementar um suporte para a construção de aplicações multimídias cooperativas sobre a rede Internet. Foi dada ênfase ao suporte de aplicações onde os participantes cooperam de forma síncrona e compartilham tanto elementos de natureza discreta (como documentos, desenhos e textos) quanto mídias contínuas (como por exemplo, voz). Foi adotado um suporte de sessão de trabalho cooperativo que permite a ordenação de mensagens respeitando o tempo de validade destas. Escolheu-se enfim uma aplicação específica para exemplificar a funcionalidade do suporte proposto usando ferramentas já existentes na rede MBone.

Os resultados que obtidos, tanto no projeto como na implementação do suporte, foram satisfatórios, comprovando as escolhas feitas em termos de algoritmos de ordenação e de sincronização de relógios. Acredita-se que no futuro novos trabalhos poderão ser desenvolvidos, na continuação deste. Em resumo, pode-se mencionar alguns aspectos de interesse que poderiam ser abordados:

Controle de Concorrência

Implementar vários mecanismos de controle de concorrência mais ou menos estritos [Fritzke95a] que poderão ser escolhidos pelos usuários da sessão cooperativa.

Mecanismos de Sincronização Intramídia

Incorporar mecanismos de sincronização intramídia, não existentes em nenhuma das ferramentas do MBone, de forma a melhorar a qualidade da apresentação das mídias contínuas.

Ferramenta whiteboard com Telepointer

Como o código da ferramenta WB não está disponível, o "*telepointer*" desenvolvido teve como única funcionalidade a simples indicação de localização de objetos na janela do WB. Um "*white board*" com "*telepointer*" foi desenvolvido no LCMI por [Fritzke95b]. Uma adaptação deste no contexto da aplicação aqui descrita pode constituir-se numa continuação deste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

- [Baldoni96] Baldoni R., Mostefaoui A., Raynal M., "*Causal Delivery of Messages with Real-Time Data in Unreliable Networks*", Journal of Real-Time Systems 10 (1996): 245-262.
- [Birman87] Birman K., Joseph T., "*Reliable Communication in the presence of failure*", ACM Transactions on Computer Systems, Vol 5 No. 1 (1987) : 47-76.
- [Braden95] Braden R., Estrin D., Jamin S., "*Resource Reservation Protocol (RSVP): Version 1 Funcional Specification* ", Internet Draft (May 1995).
- [Casner92] Casner S., Deering S., "*First IETF Internet Audiocast*", ACM SIGCOM Computer Communications Review (July 1992) : 92-97.
- [Casner94] Casner S., "*Are you on the Mbone?*", IEE Multimedia (Summer 1992) : 76-79.
- [Cosquer96] Cosquer F. J. N., Veríssimo P., "*Survey of Selected Groupware Applications and Supporting Plataforms*", Submitted for Publication, (1996), INESC, Lisboa, Portugal.
- [Deering89] Deering S., "*Host Extensions for IP Multicast*", Network Working Group Request for Comments, RFC 1112, Standford University, 1989.
- [Deering90] Deering S., Cheriton D., "*Multicast Routing in Datagram Internetworks and Extended LANs*", ACM Transactions on Computer Systems, (May 1990) : 85-111.
- [Ellis91] Ellis C. A., Gibbs S. J., Rein G. L., "*Groupware-Some Issues and Experience*", Communications of the ACM, Vol. 34, No. 1, (January 1991).
- [Floyd95] Floyd S., Jacobson V., McCanne S., "*A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framming*", ACM SIGCOM 1995.
- [Fritzke95a] Fritzke U., "*Projeto e Implementação de um Suporte para Aplicações Cooperativas do Tipo Editor Distribuído*", Dissertação de Mestrado, Curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina, (Novembro 1995), Brasil.
- [Fritzke95b] Fritzke U., Farines J.M. " A Support Platform for Distributed Editing", Cytedit-Ritos International Workshop on Groupware CRIWG'95, Lisboa, (september 1995).
- [Kopetz89] Kopetz H., Andreas D., Koza C., Mulazzani M., Schwabl W., Sentf C., Zainlinger R., "*Distributed fault-tolerant real time systems - the Mars approach*", IEE Micro (1989): 25-41.
- [Kumar95] Kumar V., "*Mbone: Interactive Multimedia on the Internet*", NewRiders Publishing, 1995.
- [Lamport78] Lamport L., "*Time clocks and the ordering of events in distributed systems*", Communications of the ACM, Vol. 21 No. 7 (July 1978) : 558-565.

- [Macedonia94] Macedonia M. R., Brutzman D. P., "MBone Provides Audio and Video Across the Internet", IEE Computer Magazine, (April 1994) : 30-36.
- [Mills89] Mills D., "Measured performance of the Network Time Protocol in the Internet System", Network Working Group Request for Comments: 1128, (October 1989).
- [Mills91] Mills D., "Internet Time Synchronization: The Network Time Protocol", IEEE Transactions on Communications, Vol 39 No. 10 (October 1991) : 1482-1493.
- [Mills92] Mills D., "Network Time Protocol (Version 3) specification, implementation and analysis", Network Working Group Report RFC-1305, (March 1992).
- [Moy94] Moy J., "Multicasting Routing Extensions for OSPF", Communications of the ACM, Vol. 37, (August 1994) : 61-66.
- [Savetz96] Savetz K., Randall N., Lepage Y., "MBone: Multicasting Tomorrow's Internet", IDG Books Worldwide, 1996.
- [Stefik87] Stefik M. et al, "Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings", Communications of the ACM, Vol. 30, No. 1, (January 1987) : 32-46.
- [Tanenbaum94] Tanenbaum A S., "Redes de Computadores", Segunda Edição Americana, Editora Campus, Brasil, 1994.
- [Veríssimo96] Veríssimo P., Rodrigues L., Casimiro A., "CesiumSpray: A Precise and Accurate Global Clock Service for Large-Scale Systems", Submitted for Publication, (1996), INESC, Lisboa, Portugal.
- [Waitzman88] Waitzman D., Partridge C., Deering S., "Distance Vector Multicasting Routing Protocol", Network Working Group Request for Comments, Internet RFC-1075, (November 1988).
- [Yavatkar92] Yavatkar R., "MCP: A Protocol for Coordination and Temporal Synchronization in Multimedia Collaborative Applications", The 12th Internacional Conference on Distributed Computing System, (1992): 606-613.
- [Zhang93] Zhang L. et al, "RSVP: A New Resource Reservation Protocol", IEE Network, (September 1993) : 8-18.