

# Avaliação do Protocolo CAN *Multicast* no Suporte a Ambientes Virtuais Colaborativos

Welsing M. Pereira<sup>1</sup>, Artur Ziviani<sup>2</sup>, Jauvane C. de Oliveira<sup>3</sup>, Paulo C. S. Vidal<sup>1</sup>

<sup>1</sup>Departamento de Engenharia de Computação - Instituto Militar de Engenharia (IME)  
Praça General Tibúrcio, 80, Praia Vermelha - Rio de Janeiro – RJ – Brasil

<sup>2</sup>Coordenação de Sistemas e Redes  
Laboratório Nacional de Computação Científica (LNCC)

<sup>3</sup>Coordenação de Ciência da Computação  
Laboratório Nacional de Computação Científica (LNCC)  
Av. Getúlio Vargas, 333 - Quitandinha - Petrópolis – RJ - Brasil

welsing@de9.ime.eb.br, ziviani@lncc.br,  
jauvane@lncc.br, vidal@de9.ime.eb.br

**Abstract.** Most advanced Collaborative Virtual Environments (CVE) architectures rely heavily in the use of network-layer multicasting. As network-layer multicasting is still not broadly available in the Internet, alternatives need to be sought after. Application-layer multicasting (ALM) has recently emerged as a good alternative, as it does not rely on network-layer multicasting. This paper presents the use of CAN Multicast protocol for support in CVE architectures. We evaluate the performance of the Multicast CAN protocol in the light of SPLINE architecture requirements.

**Resumo.** Muitas arquiteturas para Ambientes Virtuais Colaborativos (AVCs) avançadas contam fortemente com o uso da comunicação multicast na camada de rede. Entretanto, a comunicação multicast na camada de rede não está prontamente disponível na Internet. Multicast na camada da aplicação (ALM – Application Layer Multicast) tem recentemente surgido como uma boa alternativa ao multicast na camada de rede. Este artigo apresenta o uso do protocolo CAN Multicast no suporte a AVCs em grande escala. Foram realizadas simulações para avaliar o desempenho do protocolo a partir dos requisitos da arquitetura SPLINE.

## 1. Introdução

O protocolo IP *Multicast* é uma solução para a comunicação de grupos *multicast* com grande número de participantes e apresenta uma técnica de roteamento que possui eficiência na entrega de pacotes a partir de uma única origem a múltiplos destinos [Deering 1990]. Essa técnica elimina a replicação de pacotes redundantes em um mesmo enlace físico na rede. Entretanto, sua implementação na rede requer suporte aos diversos roteadores espalhados geograficamente [Diot et al 2000]. Além disso, os roteadores devem manter os estados de roteamento para cada grupo *multicast* criado, o que torna o modelo pouco escalável com relação ao número de grupos *multicast* ativos concurrentemente. Estes fatores têm limitado uma adoção abrangente do protocolo IP *Multicast* na Internet.

Um exemplo de rede que utiliza o protocolo IP *Multicast* é o Mbone (*Multicast Backbone*) [Erikson 1994]. O Mbone consiste de sub-redes (chamadas de ilhas) que oferecem suporte *multicast* ligadas entre si por meios de enlaces ponto-a-ponto usando a técnica de tunelamento. O Mbone utiliza-se de túneis para que os pacotes *multicast* atravessem as redes por meio de encapsulamento dos pacotes. Os pacotes chegam até o próximo roteador compatível que desencapsula os pacotes e os transmite ao destino final. Porém, dificuldades de configuração dos túneis bem como a administração e gerenciamento dos problemas inerentes ao protocolo IP *Multicast* fazem com que poucos Provedores de Serviços Internet (PSIs) disponibilizem conectividade *multicast* para usuários domésticos.

Existe um grande número de aplicações com diferentes exigências do ponto de vista do projeto do protocolo IP *Multicast*, como a vídeo-conferência, o ensino a distância e as salas privadas de bate-papo. Estas aplicações em geral contêm um número pequeno de participantes dentro do grupo e os grupos são criados e eliminados com certa freqüência. Quando há um grande número desses pequenos grupos, a eficiência na largura de banda e a escalabilidade oferecida pelo protocolo IP *Multicast* é reduzida pela complexidade do controle ligada à configuração e manutenção dos grupos [Pendakaris et al. 2001]. Há um crescimento no número de aplicações desses tipos e pesquisadores vêm procurando por soluções que não dependam de suporte *multicast* nos roteadores.

*Multicast* na camada de aplicação ou *Application Layer Multicast* (ALM) ou ainda *End System Multicast* (ESM) é uma alternativa que tem o potencial de resolver os problemas associados ao protocolo IP *Multicast* [Pendakaris et al. 2001] [Banerjee et al. 2002]. Apesar de não possuir o mesmo desempenho do protocolo IP *Multicast*, protocolos ALM não dependem de alterações na infra-estrutura da rede. Em vez disso, um serviço similar ao protocolo IP *multicast* é implementado a partir da colaboração entre sistemas finais através do encaminhamento de pacotes. Em um protocolo ALM todos os pacotes são transmitidos de forma distribuída, usando o roteamento ponto-a-ponto (*unicast*). Os protocolos ALM são alternativas importantes ao protocolo IP *Multicast*. Assim, vários protocolos ALM foram propostos para atender as diversas aplicações como: o ALMI [Pendakaris et al. 2001], o NICE [Banerjee et al. 2002], o NARADA [Chu et al. 2002], o Scribe [Castro et al. 2002], o CAN *Multicast* [Ratnasamy et al. 2001b], o Bayeux [Zhuang et al. 2001], o Yoid [Francis 2006] e o RMX [Chawathe 2002].

Um protocolo ALM é construído com base a atender aplicações específicas. Com isso, requisitos como o atraso fim-a-fim, a escalabilidade, a tolerância a falhas, o caminho de dados ser específico a cada origem ou ser compartilhado entre os sistemas finais devem ser considerados antes da escolha do protocolo.

Em aplicações baseadas em Ambientes Virtuais Colaborativos (AVCs) de grande escala o uso de um protocolo ALM é uma solução alternativa para redução de tráfego redundante na rede gerada pela necessidade de atualização das cópias do mundo virtual entre os participantes. A utilização da comunicação *multicast* permite melhorias na escalabilidade do ambiente virtual, pois este tipo de comunicação proporciona o aumento do número de usuários no ambiente. No entanto, sua baixa difusão acarreta na exclusão de participantes cujo PSIs não ofereçam suporte ao protocolo IP *Multicast*. Com isso, encontrar um protocolo ALM que melhor atenda às aplicações para AVCs de grande escala se torna uma importante contribuição para aqueles que têm o objetivo de

disponibilizar uma aplicação para tais sistemas e que não querem ter as dificuldades do protocolo IP *Multicast*.

O artigo [Pereira et al. 2005] apresenta uma análise comparativa entre diversos protocolos ALM existentes na literatura. Dentre os protocolos analisados, os protocolos TBCP (*Tree Building Control Protocol*) [Mathy et al. 2001], Scribe e CAN *Multicast* são adequados às aplicações AVCs de grande escala.

Assim, este artigo apresenta uma avaliação do protocolo CAN *Multicast* em diversos cenários, fornecendo suporte a arquitetura para AVC de grande escala chamada SPLINE. Os resultados obtidos demonstram que o protocolo CAN *Multicast* é uma alternativa bastante adequada no suporte a diversos AVCs de grande escala, desde que propriamente utilizado.

Este artigo está organizado da seguinte maneira. A Seção 2 apresenta a arquitetura SPLINE para ambientes virtuais colaborativos de larga escala. A Seção 3 descreve a gerência de grupos *multicast* e as aplicações dos protocolos ALMs. A Seção 4 apresenta uma síntese das principais características dos protocolos ALMs adequados às aplicações AVCs de larga escala. A Seção 5 descreve o modelo de simulação do protocolo CAN *Multicast* e os resultados de simulação estão contidos na Seção 6. A Seção 7 é dedicada às considerações finais do trabalho.

## 2. A arquitetura SPLINE para AVCs de grande escala

Ambientes virtuais colaborativos tem o objetivo de aplicar a realidade virtual em um espaço (mundo virtual) formado por uma rede de usuários permitindo que esses colaborem e compartilhem objetos entre si como se estivessem fisicamente em um mesmo local.

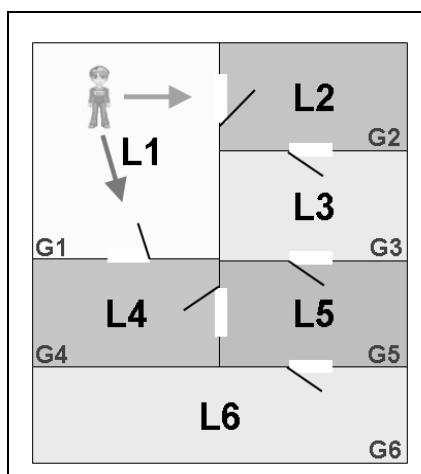
Ambientes virtuais colaborativos de grande escala são diferenciados de outras aplicações por permitirem que um número suficientemente grande de participantes compartilhem e interajam em um mesmo mundo virtual. Jogos massivos com múltiplos jogadores [Knutsson 2004], simulações distribuídas e aplicações colaborativas de grande escala são exemplos de AVCs de grande escala. Podemos citar como exemplos de arquiteturas para AVCs de grande escala: a DIVE [Carlsson e Hagsand 1993], a SPLINE [Barrus et al. 1996] e a BrickNet [Singh et al. 1995], entre outras.

Em nosso trabalho, o protocolo CAN *Multicast* é simulado no suporte a AVCs de grande escala provendo suporte a arquitetura SPLINE. A arquitetura SPLINE foi escolhida para representar o mundo virtual em nossa simulação por ser uma arquitetura simples e representar bem as arquiteturas baseadas em divisão espacial [Oliveira 2003].

A arquitetura SPLINE (*Scalable Platform for Large Interactive Networked Environments*) foi projetada pela MERL (*Mitsubishi Electric Research Laboratories*). Na arquitetura SPLINE, o mundo virtual é dividido em áreas chamadas *locales*. *Locales* podem variar de tamanho e de forma. Para cada *locale* é associado um grupo *multicast*. Cada *locale* define seu próprio sistema de coordenadas. Nesta arquitetura, os participantes que entram em um determinado *locale* recebem informação de todos os objetos pertencentes aquele *locale* e de seus vizinhos imediatos.

Para a comunicação de dados entre os usuários do mundo virtual, a arquitetura SPLINE usa o protocolo de transferência compartilhada interativa denominado ISTP (*Interactive Sharing Transfer Protocol*) [Waters 1997]. ISTP que é um protocolo híbrido

que suporta muitos modos de transportes de dados e as informações para realidade virtual são dadas por meio de cinco subprotocolos: *I-I Connection*, *Object State Transmission*, *Streaming Áudio*, *Locale-Based Communication* e *Content-Based Communication*.



**Figura 1: Modelo do mundo em SPLINE.**

A Figura 1 representa um modelo do mundo virtual na arquitetura SPLINE. Nela, as áreas geométricas definem o espaço apropriado por diferentes *locales*. As portas entre *locales* simbolizam vizinhanças entre eles. Se um membro deseja associar-se ao *locale* L1, esse membro deve assinar seu grupo *multicast* (G1) associado e os grupos *multicast* associados aos seus *locales* vizinhos (G2, G4).

### 3. Aplicações e gerência dos protocolos ALMs

*Multicast* na Camada de Aplicação é uma importante alternativa à comunicação *multicast* na camada de rede por questões de gerenciamento (configuração e controle) e por dispensar qualquer modificação na infra-estrutura básica da rede.

Em ALM, uma sobre-camada de rede é construída no topo dos serviços de redes disponíveis e os pacotes ponto-a-ponto são encaminhados entre sistemas finais através de uma aplicação. O modelo é organizado de forma que cada sistema final participante (membro) é responsável por entregar pacotes de dados a um subconjunto de participantes que estão ligados a ele por meio de alguma regra. A esse subconjunto de participantes chamamos de participantes vizinhos.

Os grupos *multicast*, em ALM, podem ser gerenciados de forma centralizada ou distribuída. Na forma centralizada, um ou mais sistemas finais, que podem ou não participar do grupo *multicast*, são responsáveis por manter estados dos membros no grupo além de identificar e reparar possíveis partições causadas por uma saída ou queda de um membro dentro do grupo.

Na forma distribuída, os próprios membros participantes são responsáveis por manter seus estados e também pela identificação e reparo de possíveis partições dentro do grupo. A forma da gerência de grupo pode limitar a quantidade de usuários no sistema. Em uma gerência centralizada, a quantidade limitada de sistemas finais no gerenciamento limita a

escalabilidade do sistema pela capacidade computacional desses sistemas finais além de introduzir pontos centrais de falha. Uma gerência distribuída favorece a escalabilidade do sistema (membros são responsáveis por manter estados de outros poucos membros do grupo) e pontos de falhas são limitados a poucos membros no grupo.

Uma aplicação de grande escala necessita de uma abordagem distribuída na gerência de grupo. Os protocolos que implementam uma gerência de grupo distribuída são: CAN *Multicast*, Scribe, TBCP e Yoid. Os protocolos ALMI (*Application Level Multicast Infrastructure*) [Pendakaris et al. 2001], CoopNet [Padmanabhan et al. 2002], AMCast [Shi e Turner 2002], RMX [Chawathe et al. 2002] e HBM (*Host-Based Multicast*) [Roca e El-Sayed 2001] são exemplos de protocolos que utilizam a gerência de grupo centralizada.

Os protocolos ALM são construídos com base nas aplicações a que eles se destinam e suas características são definidas de acordo com as necessidades das aplicações.

As aplicações para fluxo de áudio/vídeo em geral envolvem uma única origem para distribuição de mídia e uma grande quantidade de clientes receptores. Tanto mídias de fluxo em tempo real quanto mídias de fluxo sob demanda são exemplos dessas aplicações. AMCast [Shi e Turner 2002], NICE [Banerjee et al. 2002], ZIGZAG [Tran et al. 2003] e Bayeux [Zhuang et al. 2001] são protocolos construídos com base nessas aplicações. Esses protocolos têm em comum a utilização do atraso e ou banda como métricas para construção da topologia de dados e a transmissão de dados é baseada em uma única fonte com destino a múltiplos receptores.

Aplicações para áudio-conferência e ou vídeo-conferência pertencem a uma outra classe de aplicações. Em geral são constituídas a partir de pequenos e médios grupos de participantes que interagem em uma sessão de conferência. Nessas aplicações, um alto grau de interatividade é exigido e cada participante tem o potencial de enviar e receber dados, ou seja, nessas aplicações contamos com múltiplas fontes e múltiplos destinos. ALMI [Pendakaris et al. 2001], HBM [Roca e El-Sayed 2001], HostCast [Li e Mohapatra] e NARADA [Chu et al. 2002] são protocolos para estas aplicações.

Aplicações para simulações distribuídas de grande escala formam uma classe constituída por um grande número de participantes, onde cada participante é uma origem de dados. Essas aplicações requerem transmissão de mensagens muitos-para-muitos, uma vez que seus participantes são livres para trocas de uma grande quantidade de informações de atualização simultânea. Jogos massivos com múltiplos participantes e aplicações para AVC de grande escala são exemplos dessas aplicações. Nessas aplicações o atraso é considerado a principal métrica na topologia de dados. Os protocolos CAN *Multicast*, Scribe , TBCP e Yoid são construídos para esses fins.

#### 4. Protocolos ALM no suporte a AVCs

Os ambientes virtuais colaborativos de grande escala possuem características próprias como tipo de transmissão muitos-para-muitos, gerência distribuída de grupo *multicast* e o atraso fim-a-fim como principal métrica na construção da árvore ALM. Com base nessas características, o artigo de Pereira et al. [Pereira et al. 2005] realizou uma análise comparativa entre 23 (vinte e três) protocolos ALM e concluiu que os protocolos TBCP, Scribe e CAN *Multicast* são adequados as aplicações AVCs de grande escala.

O protocolo TBCP (*Tree Building Control Protocol*) [Mathy et al. 2001] foi projetado para reduzir o tempo de convergência na construção de uma “boa” árvore a partir do conhecimento de poucos participantes. Este protocolo limita-se a posicionar novos membros em uma localização ótima em tempo de entrada no grupo e assumindo cada nó tem um número limitado de filhos para servir.

A entrada de um novo membro no grupo requer trocas de mensagens entre o novo membro e um candidato a nó pai do novo membro. Na entrada em um grupo, o novo membro contata o nó pai candidato e o nó pai candidato replica ao novo membro uma lista de seus nós filhos. Baseado em alguma função de custo, o novo membro avalia o nó pai assim como os nós filhos para selecionar seu melhor ponto de entrada no grupo.

O protocolo TBCP é limitado, pois não possui um subconjunto de funções necessárias a sua implementação em uma ambiente de simulação, como por exemplo: o procedimento de entrada e saída de um membro no grupo *multicast*.

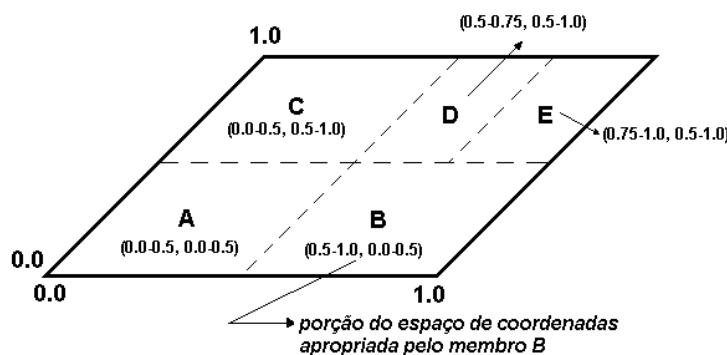
Scribe [Castro et al. 2002] é uma infra-estrutura ALM escalável construída no topo de uma estrutura *Pastry*. *Pastry* [Rowston e Druschel 2001] é uma estrutura de sobre-camada P2P completamente descentralizada onde cada nó participante tem igual responsabilidade. A estrutura *Pastry* possui uma distribuição uniforme de seus nós baseado em uma tabela *hash* segura, assegurando assim uma população regular em seu espaço.

No protocolo Scribe, qualquer nó pode criar um grupo e outros podem unir-se a esse grupo e enviar mensagens *multicast* a ele (desde que tenham uma credencial apropriada ao grupo). Um sistema Scribe consiste de uma rede de nós *Pastry* [Castro et al. 2002]. Cada nó executa o programa de aplicação Scribe.

O protocolo Scribe cria uma árvore *multicast*, enraizada em um RP (*Redezvous Point*), para disseminar mensagens *multicast* no grupo. Sua árvore *multicast* é criada usando um esquema similar ao RPF (*Reverse Path Forwarding*). A árvore é formada pela adição de rotas *Pastry* de cada membro do grupo ao RP. Operações de entrada no grupo são gerenciadas de maneira descentralizada para suportar muitos e dinâmicos participantes.

CAN's (*Content-Addressable Networks*) [Ratnasamy et al. 2001a] são estruturas de sobre-camada de rede par-a-par que provê funcionalidades de tabelas *hash* em grande escala. Em CAN, os nós são divididos em um espaço de coordenadas, e cada nó é dono de uma porção desse espaço de coordenadas. A Figura 2 ilustra uma CAN de dimensão 2 ocupada por 5 membros. Mensagens em CAN são roteadas através do espaço seguindo um caminho dentro do espaço cartesiano .

A extensão de CAN para oferecer *multicast* pela camada de aplicação, resultou no protocolo CAN *Multicast* [Ratnasamy et al. 2001b]. A comunicação *multicast* é realizado através de um mecanismo especial de inundação (*flooding*), onde heurísticas são utilizadas para minimizar o número de pacotes duplicados encaminhados.



**FONTE: (RATNASAMY, 2001a)**

**Figura 2: Exemplo de um espaço de coordenadas de dimensão = 2 com 5 membros.**

A partir do estudo de vinte e três protocolos ALM e dos três protocolos ALM adequados aos AVCs de grande escala, podemos afirmar que o protocolo TBCP não possui um subconjunto de funções necessárias a sua implementação em um ambiente de simulação e que, por outro lado, os protocolos CAN *Multicast* e Scribe possuem este subconjunto. Entretanto, em uma primeira etapa o protocolo CAN *Multicast* foi escolhido para implementação em um ambiente de simulação pela vasta documentação disponível sobre este protocolo, facilitando seu desenvolvimento no simulador. Em um trabalho futuro, a implementação do protocolo Scribe permitirá sua comparação com o protocolo CAN *Multicast* e então poderíamos identificar qual o protocolo mais adequado para AVCs de grande escala e em que condições isso deve ocorrer.

## 5. Avaliação do Protocolo CAN *Multicast*

Esta Seção apresenta uma avaliação através de simulação do protocolo CAN *Multicast* integrado com a arquitetura SPLINE. A métrica utilizada é o atraso (latência) nas interações entre os usuários e o mundo virtual.

Escolhemos a arquitetura SPLINE para representar o mundo virtual em nossa simulação devido sua simplicidade de implementação e por representar bem as arquiteturas baseadas em divisão espacial.

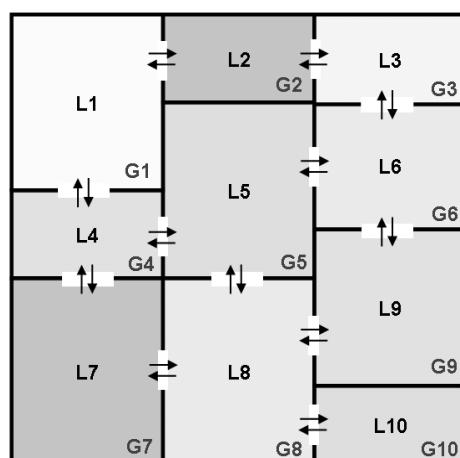
A latência é um dos mais importantes parâmetros em aplicações AVCs. A latência impacta diretamente o realismo de experiências AVCs por determinar quando as informações AVCs recebidas da rede são atualizadas [Singhal e Zyda 1999].

Tipicamente valores da latência para manter interações em tempo real variam entre 40 e 300ms segundo Diot e Gautier [Diot 1999]. O artigo de Cheshire [Cheshire 1996] sugere que a latência ida e volta para interações em rede em geral deve ser de no máximo 100ms. Enquanto os trabalhos de Wloka [Wloka 1995] e de Diot [Diot 1999] admitem 100ms de atraso em um único sentido. No entanto, a latência máxima permitida entre usuários de AVCs vai depender da aplicação.

O protocolo ALM CAN *Multicast* é um subconjunto de funções da arquitetura SPLINE foram implementados no simulador de redes J-sim [Tyan 2002]. Os núcleos dos protocolos foram escritos em Java, enquanto os cenários de simulação foram escritos na linguagem script *Tcl*.

## 5.1. Modelo de Simulação

Nossa simulação conta com um modelo do mundo dividido espacialmente em 10 áreas, cada uma representando um *locale*, segundo a arquitetura SPLINE. Cada *locale* no mundo possui um ou mais *locales* vizinhos e, de acordo com a arquitetura, cada *locale* é conhedor de seus vizinhos. A dimensão de cada *locale* não é importante em nossa simulação, assim sua forma é representada na Figura 3. A movimentação de membros entre *locales* é realizada de forma probabilística onde cada membro do mundo possui uma probabilidade de entrada, de movimentação (entre *locales* vizinhos) e de saída do mundo virtual.



**Figura 3: Modelo do mundo. Setas indicam vizinhança entre os *locales*. Onde, para cada *Locale*  $L_i$ , ( $i$  entre 1-10), tem associado o grupo CAN Multicast  $G_i$ .**

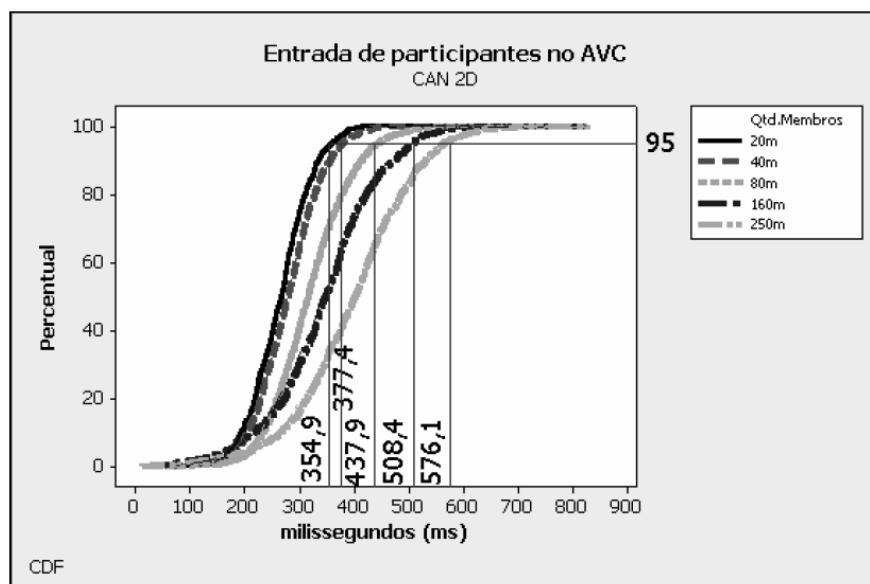
As simulações foram executadas usando CANs de 2 e 3 dimensões para mundos virtuais com: 20, 40, 80, 160 e 250 participantes. Nas simulações, os participantes podem acessar o mundo, deixar o mundo e mover-se no mundo a cada 1s. Mensagens *multicast* de atualização de estado foram enviadas por cada participante no mundo a cada 100ms, ou seja, 10 mensagens de atualização por segundo por cada participante. Cada experimento realizado envolveu 300s de simulação. A topologia utilizada compreendeu 1000 nós representando sistemas finais.

A topologia utilizada nas simulações foi baseada no modelo de Waxman [Waxman 1988] e construída a partir de um gerador de topologia chamado BRITE [Medina et al. 2001]. A latência do enlace entre pares de nós variou na faixa de 1-20ms. Foram realizados cinco repetições em cenários do mundo virtual com 20 e 40 participantes. No entanto, devido a grande quantidade de tempo requerido pelo simulador, nos cenários com 80, 160 e 250 participantes menos repetições foram efetuadas para esses experimentos.

## 6. Análise dos resultados

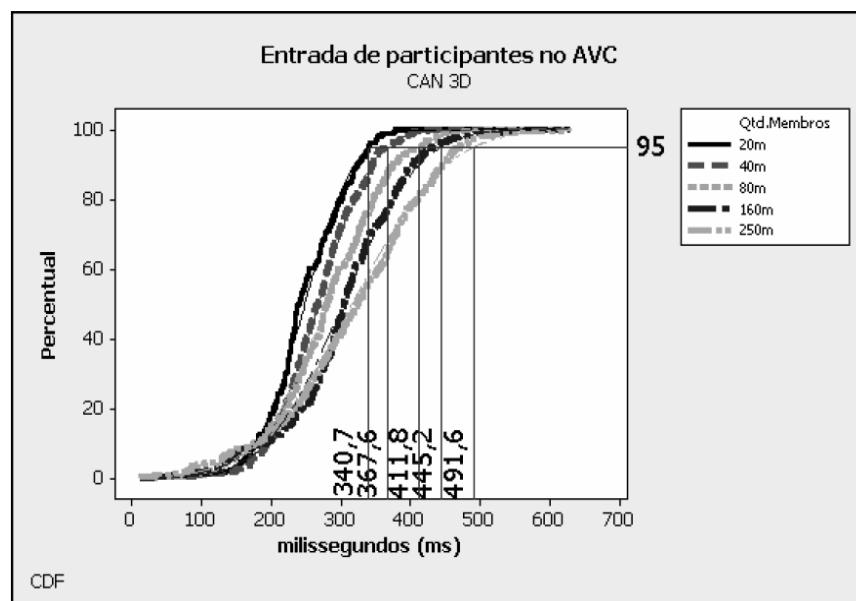
### 6.1 Entrada no Mundo Virtual

Para entrar em um mundo virtual um usuário deve trocar mensagens de controle com um servidor de *locales*. O servidor de *locales* é responsável por armazenar informações sobre o *locale* (qual o grupo *multicast* associado, quem são seus *locales* vizinhos, quais os objetos pertencentes ao *locale*). Recebida as informações do mundo virtual, o usuário então deve assinar os grupos *multicast* associado ao *locale* pretendido e de seus *locales* vizinhos.



**Figura 4:CDF do overhead para entrada de membros no mundo virtual em CANs com 2 dimensões.**

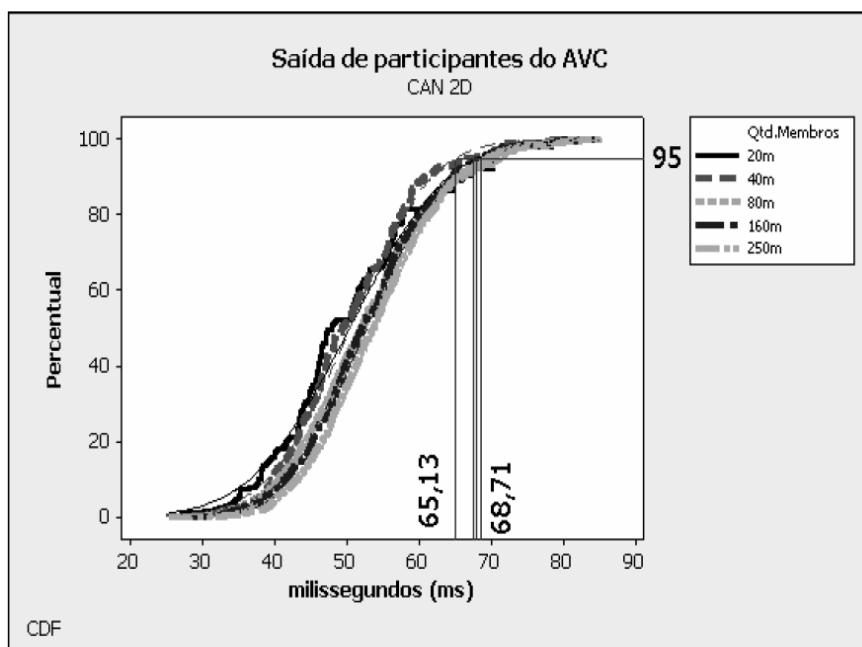
As funções de distribuição acumulada (*Cumulative Distribution Function* - CDF) das Figuras 4 e 5 representam o custo gasto por um usuário para entrar no AVC a partir da infra-estrutura CAN. Tanto em uma CAN 2D quanto em uma CAN 3D o custo é relativamente alto de acordo com os valores indicados para aplicações em tempo real segundo [Diot 1999], [Cheshire 1996] e [Wloka 1995] para manter o realismo do sistema. No entanto, para aplicações AVC em tempo real cujo custo relativo à entrada de membros não interfiram no realismo do mundo, o emprego do protocolo CAN *Multicast* pode se tornar uma boa alternativa.



**Figura 5:CDF do overhead para entrada de membros no mundo virtual em CANs com 3 dimensões.**

## 6.2 Saída do Mundo Virtual

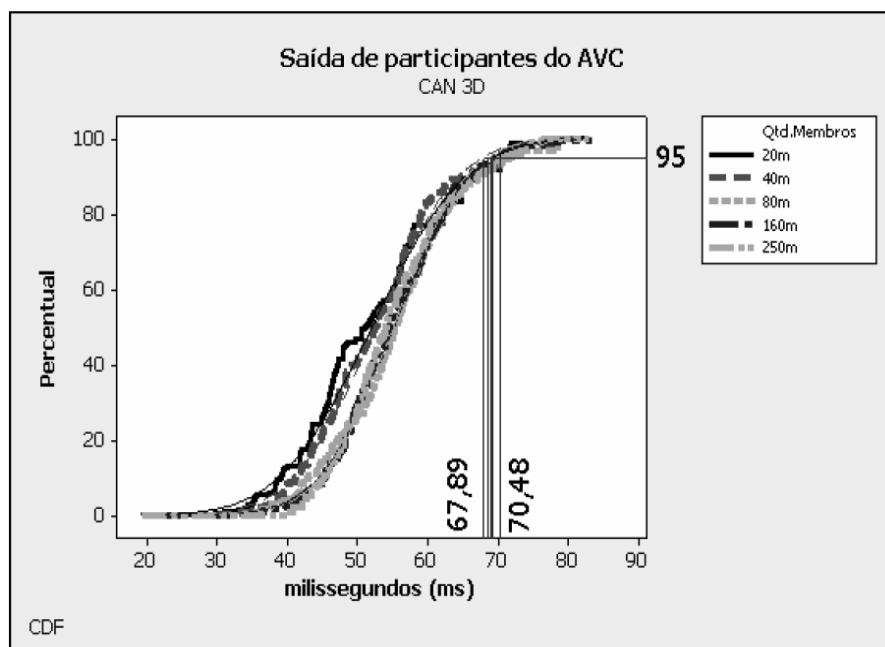
Um usuário que deseja sair do mundo virtual deve retirar suas assinaturas dos grupos associados ao *locale* no qual estava presente e aos grupos associados aos seus *locales* vizinhos.



**Figura 6: CDF do overhead para saída de membros do mundo virtual em CANs com 2 dimensões.**

As CDFs das Figuras 6 e 7 exibem o custo no 95º percentil para a saída de usuários do mundo virtual empregando CANs de 2 e 3 dimensões respectivamente. As CDFs mostram que 95% dos membros deixam o mundo virtual em até 80ms. Este tempo não sofre influência da quantidade de membros no mundo virtual e nem da dimensão CAN.

A saída de um membro do mundo virtual não interfere no realismo dos AVCs, pois seu custo envolvido está dentro da faixa de tempo sugerida por [Diot 1999], [Cheshire 1996] e [Wloka 1995] à manutenção do realismo nas aplicações de tempo real.



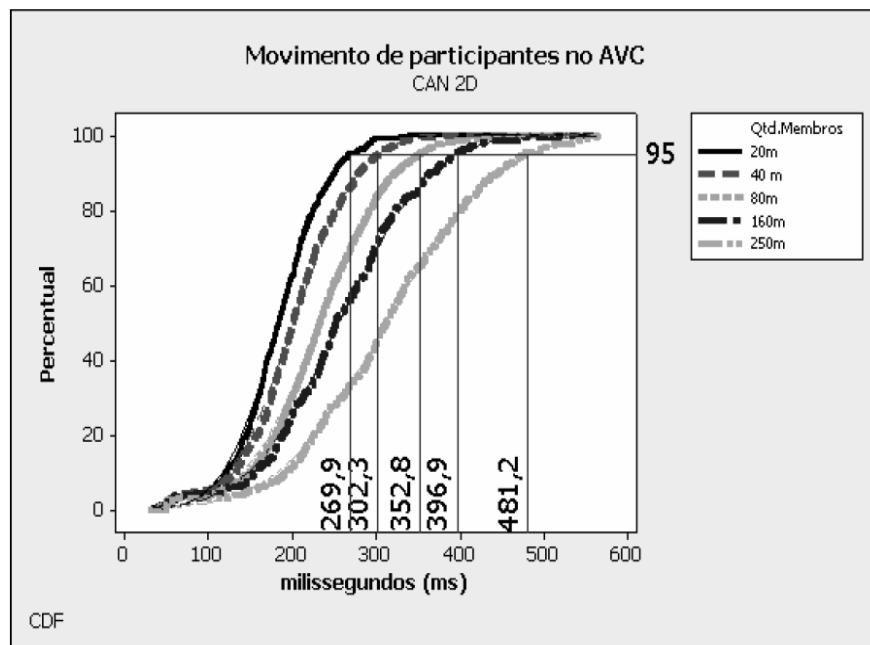
**Figura 7:CDF do overhead para saída de membros do mundo virtual em CANs com 3 dimensões.**

### 6.3 Movimento no Mundo Virtual

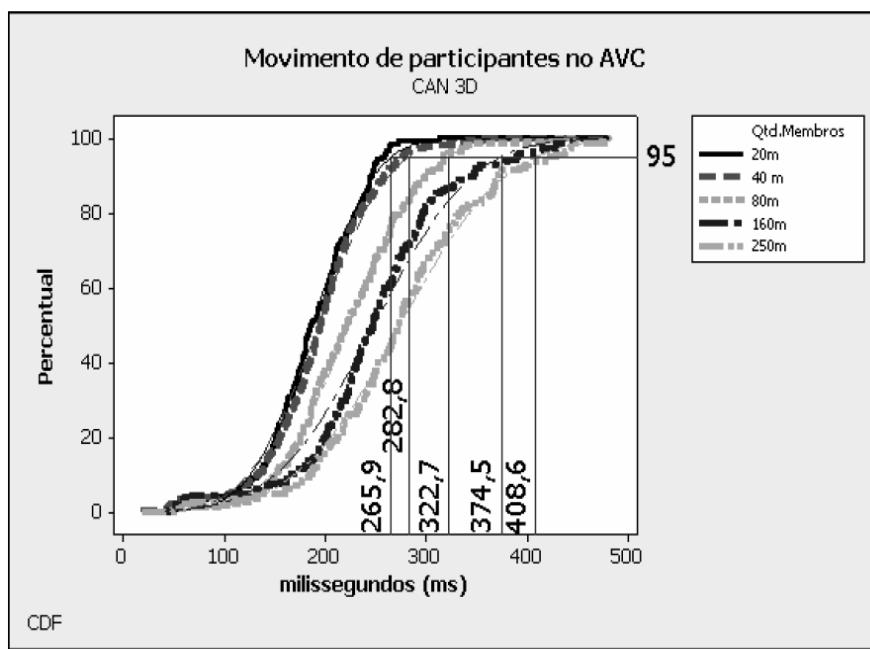
Um usuário que deseja movimentar-se entre *locales* no mundo virtual deve assinar grupos *multicast* de novos *locales* e retirar sua assinatura de outros grupos.

As CDFs das Figuras 8 e 9 exibem o custo no 95º percentil para a movimentação de um participante entre *locales* no mundo virtual empregando CANs 2D e CANs 3D, respectivamente. As CDFs mostram que o tempo gasto no movimento por um membro entre um antigo *locale* e um novo *locale* no mundo virtual é relativamente alto comparado ao custo sugerido para aplicações em tempo real segundo [Diot 1999], [Cheshire 1996] e [Wloka 1995].

Entretanto, os membros no mundo virtual, segundo a arquitetura SPLINE, assinam os grupos *multicast* associados aos *locales* vizinhos. Assim, membros do mundo virtual fazem movimentos para *locales* cujos grupos *multicast* já estão assinados por ele. Logo, a movimentação de membros no mundo virtual, em geral, não interfere no realismo do mundo virtual. Os custos associados as CDFs das Figuras 8 e 9 são custos relacionados a assinaturas de grupos *multicast* associados aos *locales* vizinhos e não propriamente ao novo *locale*.



**Figura 8: CDF do overhead para movimento de membros do mundo virtual em CANs com 2 dimensões.**



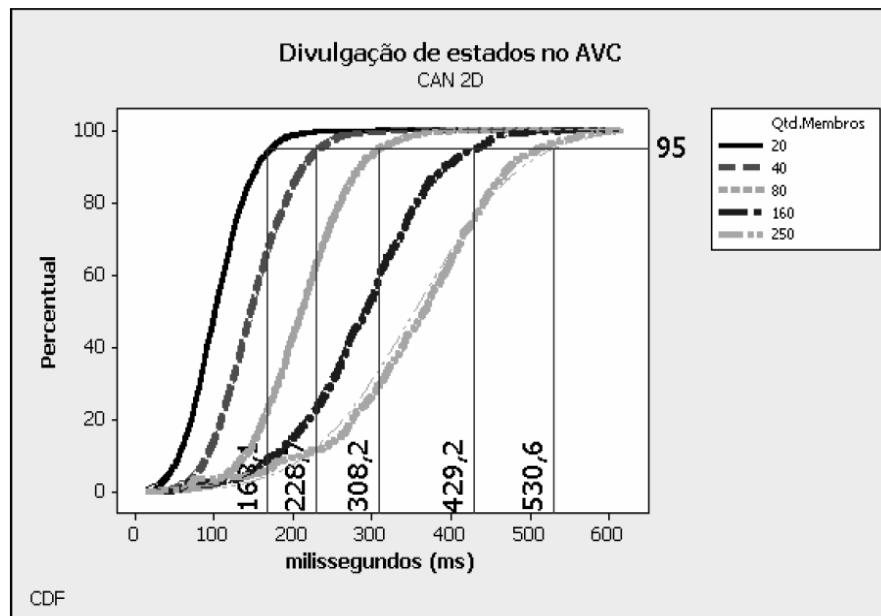
**Figura 9: CDF do overhead para movimento de membros do mundo virtual em CANs com 3 dimensões.**

#### 6.4 Divulgação de Estados entre Membros no mundo Virtual

Participantes em um mundo virtual devem periodicamente divulgar seu estado a outros participantes no mundo por meio de seus grupos *multicast*.

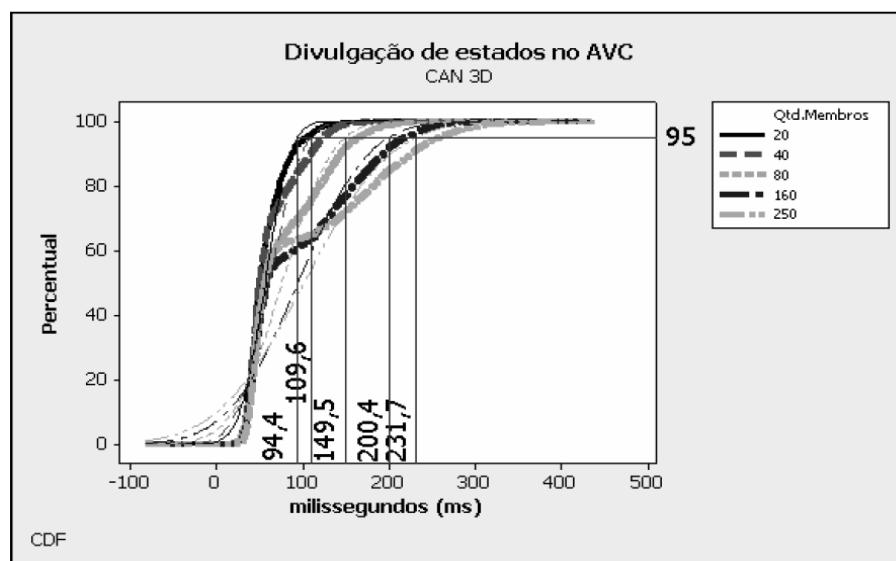
As CDFs das Figuras 10 e 11 exibem o custo no 95º percentil para envio de mensagens de atualização de estado desde seu envio por um participante da origem até a chegada ao último participante interessado. O atraso medido em uma CAN 2D para 20 e 40 participantes no mundo está dentro da faixa tolerada para manutenção do realismo segundo [Diot 1999]. Entretanto, se o número de participantes no mundo cresce e ultrapassa 80 participantes no mundo, o realismo pode estar sendo comprometido segundo a faixa de tempo sugerida por [Diot 1999].

O artigo de Ratnasamy et al [Ratnasamy 2001b] provê algumas alternativas para melhorias no protocolo CAN. Uma dessas melhorias trata o aumento na dimensão CAN. O aumento da dimensão CAN impacta na diminuição do tamanho no caminho de roteamento na  $O(d(n^{1/d}))$ .



**Figura 10: CDF do overhead para divulgação de estados entre membros no mundo virtual em CANs com 2 dimensões.**

A CDF da Figura 11 exibe os resultados do custo no 95º percentil para divulgação de estados no mundo virtual utilizando uma CAN 3D. Esses resultados mostram que mesmo com 250 participantes interagindo no mundo virtual, o tempo de divulgação de estados está dentro da faixa de tempo sugerido por [Diot 1999] para manutenção do realismo em aplicações de tempo real como AVCs. Além disso, a CDF da Figura 11 mostra que 60% das mensagens dada uma origem são entregues ao último participante interessado em menos que 100ms.



**Figura 11: CDF do overhead para divulgação de estados entre membros no mundo virtual em CANs com 3 dimensões.**

## 6.5 Resumo das Ações no Mundo Virtual

A Tabela 1 mostra que o tempo gasto para um participante deixar o mundo virtual é menor ou igual a 70ms e independe da quantidade de participantes no mundo virtual e do tamanho da dimensão CAN. Enquanto, para as demais ações, a Tabela 1 mostra que com o aumento da quantidade de participantes no mundo virtual, dada uma mesma dimensão CAN, o tempo gasto para cada ação aumenta. Por outro lado, dada uma mesma quantidade de participantes o tempo gasto para cada ação diminui com o aumento da dimensão CAN.

**Tabela 1 - Overhead das ações no Mundo Virtual.**

Quantidade de participantes no Mundo	Overhead para Entrada no Mundo no 95º ponto percentual (ms)		Overhead para Saída do Mundo no 95º ponto percentual (ms)		Overhead para Movimento no Mundo no 95º ponto percentual (ms)		Overhead para atualização de estados no Mundo no 95º ponto percentual (ms)	
	2D	3D	2D	3D	2D	3D	2D	3D
20	355	341	68	69	270	265	168	95
40	377	368	65	68	302	283	228	110
80	438	412	69	70	352	322	308	150
160	508	445	68	69	397	374	430	200
250	576	492	69	69	481	409	531	231

**Legenda:**  Valores dentro da faixa de tempo (40ms e 300ms) sugerida por DIOT (1999) para manutenção do realismo em aplicações de tempo real.

A Tabela 1 também mostra as ações com suas quantidades de participantes que estão dentro da faixa de tempo sugerida por [Diot 1999] para manutenção do realismo no mundo virtual.

O *overhead* para atualização de estados dos membros no mundo é um fator de muita importância para a manutenção do realismo no mundo virtual. A partir da Tabela 1 podemos concluir que o protocolo CAN Multicast 3D no suporte a AVCs de grande escala é um protocolo que garante o realismo do sistema dada uma quantidade de 250 participantes interagindo no mundo virtual.

## 7. Considerações Finais

Este artigo avaliou o desempenho do protocolo CAN *Multicast* no suporte a AVCs de grande escala construídas sob a arquitetura SPLINE em diversas configurações de participantes no mundo virtual assim como na variação do tamanho da dimensão CAN nessas configurações. A quantidade de participantes no mundo virtual variou entre 20 e 250 membros e o tamanho da dimensão CAN entre 2 e 3.

Resultados da simulação mostraram que o suporte do protocolo CAN *Multicast* a AVCs de grande escala podem nos levar a altos custos (*overhead*) na entrada de um membro no mundo virtual. Os altos custos devem principalmente ao fato de que o espaço virtual de coordenadas CAN é particionado entre membros sem qualquer relacionamento com a topologia física da rede. Para resolver esse problema Ratnasamy et al. [Ratnasamy 2002] provê um mecanismo da construção da árvore ALM CAN onde segundo uma métrica (o atraso), membros próximos na topologia física fiquem também próximos na topologia lógica.

Outra medida importante considerada em nossa simulação foi o tempo levado em divulgações de mensagens de estado entre membros no mundo virtual. Resultados mostraram que o tamanho da dimensão CAN é um parâmetro importante a ser considerado no suporte a AVCs de grande Escala. O custo (*overhead*) medido para divulgação de mensagens de estado dado um membro origem e sua recepção pelo último membro interessado em receber a mensagem em uma CAN 2D mostrou que acima de 80 participantes no mundo virtual o realismo do AVC poderia estar sendo comprometido, enquanto em uma CAN 3D a colaboração entre 250 participantes no mundo virtual resulta em um custo de divulgação de estados dentro da faixa de tempo (40ms a 300ms) sugerida por [Diot 1999] adequada para manutenção do realismo em aplicações de tempo real.

Os resultados avaliados nas simulações determinam que o protocolo CAN *Multicast* é uma boa alternativa no suporte a diversos Ambientes Virtuais Colaborativos de grande escala, sendo potencialmente vantajoso considerar dimensões maiores de CAN.

Dando continuidade ao trabalho desenvolvido, podemos citar como um trabalho futuro, a implementação do protocolo Scribe em um ambiente de simulação no suporte a AVCs de grande escala. Esta implementação permitirá uma comparação entre o protocolo CAN *Multicast* e o protocolo Scribe. Assim, poderíamos identificar qual o protocolo mais adequado para AVCs de grande escala e em que condições isso deve ocorrer.

## Referências Bibliográficas

- Banerjee, S.; Battacharjee, B.; Kommareddy, C. (2002) "Scalable Application Layer Multicast", ACM conference on applications, technologies, architectures, and protocols for computer communications (SIGCOMM'02), p. 205-217.
- Barrus, J. W.; Waters, R. C.; Anderson, D.B. (1996) "Locales: Supporting large multi-user virtual environments", IEEE Computer Graphics and Applications, 16(6), p. 50-57.
- Carlsson, C.; Hagsand, O. (1993) "DIVE – A multi-user virtual reality system", IEEE Virtual reality Annual International Symposium, p. 394-400.
- Castro, M.; Druschel, P.; Kermarrec, A. M. et al. (2002) "Scribe: A large-scale and decentralized application-level multicast infrastructure", IEEE Journal on Selected Areas in Communication (JSAC), 20(8), p. 1489-1499.
- Chawathe, Y.; McCanne, S.; Brewer, E.A. (2000) "RMX: Reliable Multicast for Heterogeneous Networks", IEEE INFOCOM 2000, v. 2, p. 795-804.
- Cheshire, S. (1996) "Latency and the quest for interactivity". A white paper commissioned by Volpe Welty Asset Management, L.L.C., for the synchronous person-to-person interactive computing environments meeting, San Francisco.
- Chu, Y.; Rao, S.G.; Zhang, H. (2000) "A Case for End System Multicast", Measurement and Modeling of Computer Systems, p. 1-12.
- Deering, S.; Cheriton, D. (1990) "Multicast Routing in Datagram Internetworks and Extended LANs". ACM Transactions on Computer Systems, v. 8, n. 2, p. 85-110, maio.
- Diot, C.; Gautier, L. (1999) "A distributed architecture for multiplayer interactive applications on the Internet", IEEE Network, 13(4), p. 6-15.
- Diot, C.; Levine, B.N.; Lyles, B. et al. (2000) "Deployment issues for the IP multicast Service and Architecture", IEEE Network, v. 14, n. 1, p. 78-88, janeiro/fevereiro.
- Erikson, H. (1994) "MBONE: The Multicast Backbone", Communications of the ACM, v. 37, n. 8, p. 54-60.
- Francis, P. (1999) "Yoid: Extending the Multicast Internet Architecture". Disponível em: <<http://www.aciri.org/yoid>>. Acesso em: 21 julho de 2006.
- Knutsson, B.; Lu, H.; Xu, W. (2004) "Peer-to-Peer Support for Massively Multiplayer Games", IEEE INFOCOM.
- Li, Z.; Mohapatra, P. (2003) "Hostcast: A new overlay multicasting protocol", IEEE International Conference on Communications (ICC 2003).
- Mathy, L.; Canonico, R. e Hutchison, D. (2001) "An Overlay Tree Building Control Protocol", 3<sup>rd</sup> International Workshop on Networked Group Communication, (NGC'01), p. 78-87.
- Medina, A., Lakhina, A., Matta, I. et al. (2001) "BRITE: An Approach to Universal Topology Generation". International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems- MASCOTS'01.
- Oliveira, J. C.; Georganas, N. D. (2003) "VELVET- An Adaptive Hybrid Architecture for VEry Large Virtual EnvironmenTs ". Presence: Teleoperators and Virtual Environments, v. 12, n. 6, p. 555-580.

- Padmanabhan, V.; Wang, H.; Chou, P. et al. (2002) "Distributing streaming media content using cooperative networking", NOSSDAV'02.
- Pendakaris, D.; Shi, S.; Verma, D. et al. (2001) "ALMI: An Application Level Multicast Infrastructure", 3<sup>rd</sup> Usenix Symposium on Internet Technologies and Systems, p. 49-60.
- Pereira, W.M.; Vidal, P.C.S.; Ziviani, A. e Oliveira, J.C. (2005) "Emprego de Protocolos Multicast na Camada de Aplicação no Suporte a Ambientes Virtuais Colaborativos", Workshop de Teses e Dissertações, XI Simpósio Brasileiro de Sistemas Multimídia e Web – WebMídia'2005.
- Ratnasamy, S.; Francis, P.; Handley, M.; Karp, R. e Shenker, S. (2001a) "A Scalable Content-Addressable Network", SIGCOMM'01, p. 27-31.
- Ratnasamy, S.; Handley, M.; Karp, R. e Shenker, S. (2001b) "Application-level Multicast using Content-Addressable Networks", International Workshop on Networked Group Communication, p. 14-29.
- Ratnasamy, S.; Handley, M.; Karp, R. e Shenker, S. (2002) "Topologically-aware overlay construction and server selection", IEEE INFOCOM '02.
- Roca, V.; El-Sayed, A. (2001) "A host-based Multicast (HBM) Solution for Group Communications", IEEE International Conference on Networking (ICN'01), p. 610-619.
- Rowstron, A.; Druschel, P. (2001) "Pastry: Scalable, decentralized object location, and routing for large-scale Peer-to-Peer systems". International Conference on Distributed Systems Platforms (IFIP/ACM), p. 329-350.
- Shi, S. Y.; Turner, J. S. (2002) "Routing in Overlay Multicasting Networks", IEEE INFOCOM 2002, v. 3, p. 1200-1208.
- Singh, G.; Serra, L.; Png, W. et al. (1995) "BrickNet: Sharing Object Behaviors on the Net", Virtual Reality Annual International Symposium (VRAIS'95), p. 19-25. IEEE Computer Society Press, março.
- Singhal, S., Zyda, M. (1999) "Networked Virtual Environments – Design and Implementation", Addison Wesley, 330 p., ISBN 0-201-32557-8.
- Tyan, H. (2002) "Design, realization and evaluation of a component-base compositional software architecture for network simulation", Tese de Doutorado, Graduate School of The Ohio State University.
- Tran, D.A.; Hua, K.A.; Do, T. (2003) "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming", IEEE INFOCOM, v. 2, p. 1283-1292.
- Waters, R.C.; Anderson, D. B.; Shewenke, D. L. (1997) "The Interactive Sharing Transfer Protocol Version 1.0", MERL Technical Report n. TR-97-10.
- Waxman, B. M. (1988) "Routing of multipoint connections", IEEE Journal on Selected Areas in Communications, 6:1617-1622.
- Wloka, M. (1995) "Lag in multiprocessor VR", Presence: Teleoperators and Virtual Environments, 4(1), p. 50-63.
- Zhuang, S. Q.; Zhao, B. Y.; Joseph, A.D. et al. (2001) "Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination", ACM NOSSDAV'01, p. 11-20.