

APLICAÇÃO DE TÉCNICAS DE TOLERÂNCIA A FALHAS EM UM SISTEMA DISTRIBUÍDO PARA A EDUCAÇÃO A DISTÂNCIA.

Berto de Tácio P. GOMES (1); Omar Andrés C. CORTÊS (2); Rafael Fernandes LOPES (3)

(1) Centro Federal de Educação Tecnológica do Maranhão – CEFET-MA, Avenida Getulio Vargas s/nº - Monte Castelo, CEP 65.030-000, fone/fax: (98) 3218-9000 e-mail: bertodetacio@gmail.com

(2) Centro Federal de Educação Tecnológica do Maranhão – CEFET-MA, e-mail: omar@cefet-ma.br

(3) Centro Federal de Educação Tecnológica do Maranhão – CEFET-MA, e-mail: rafaelf@cefet-ma.br

RESUMO

O avanço tecnológico das redes de computadores e da Internet tem levado a Educação a Distância a utilizar uma série de sistemas distribuídos, entre eles o *Chat*. Apesar de sua larga utilização, o *Chat* apresenta o problema de falta de coordenação da troca de mensagens, o que tem dificultado o uso dessa ferramenta em cursos a distância. O problema ocorre quando vários usuários enviam suas mensagens ao mesmo tempo, ignorando o fato de que é difícil acompanhar o que todos escreveram de forma eficiente, quando o volume de mensagens é muito grande. Nessa situação, aumenta o esforço cognitivo para fazer as ligações coesivas entre as mensagens.

Alguns sistemas distribuídos inovadores de *Chat* surgiram com o objetivo de solucionar o problema de falta de coordenação em bate-papos, no entanto esses sistemas não têm se preocupado com técnicas que os permitam continuar oferecendo seus serviços diante da presença de falhas de hardware ou de software.

O objetivo deste trabalho é mostrar técnicas de tolerância a falhas que foram aplicadas no sistema *SynchroTalk*. Este é um sistema de *Chat* elaborado para a Educação a Distância, e possui mecanismos de coordenação da troca de mensagens para auxiliar orientadores de ensino-aprendizagem na tarefa de coordenar conversações. No entanto esses mecanismos são passíveis de falhas, que são inevitáveis de acontecer em qualquer sistema distribuído. A finalidade da aplicação dessas técnicas é permitir que os mecanismos de coordenação do *SynchroTalk* possam minimizar as consequências provocadas por essas falhas, de forma que este possa continuar a oferecer seus serviços de troca de mensagens coordenadas mesmo após a ocorrência de falhas na comunicação entre usuários.

Palavras-chave: Educação a Distância, *Chat*, coordenação, sistemas distribuídos e tolerância a falhas..

1. INTRODUÇÃO

A Educação a Distância (ou EaD) vem se tornando uma modalidade de ensino bastante utilizada nos últimos anos para aumentar a oferta de ensino em várias áreas, que abrangem desde treinamentos profissionais e cursos técnicos até cursos de graduação total ou parcialmente a distância. Esta modalidade tem se tornado uma alternativa viável para expandir o ensino e possibilitar que comunidades que antes não tinham o acesso à educação pelo fato de estarem longe dos grandes centros urbanos, o tenham graças a recursos tecnológicos audiovisuais que possibilitam a comunicação entre aqueles que necessitam do conhecimento e os que potencialmente estão habilitados a oferecê-lo, mas que por motivos geográficos e/ou socioeconômicos estão impossibilitados de comparecerem de maneira físico-presencial. No entanto, até mesmo os grandes centros urbanos fazem utilização da modalidade de ensino à distância, visto que em grande parte das vezes, os profissionais responsáveis por oferecer o treinamento pertencem a uma outra região metropolitana ou país.

Uma das grandes vantagens da utilização da Educação à distância é a possibilidade de redução de custos, tais como os provenientes do deslocamento de profissionais. É preciso ressaltar que em muitas áreas de conhecimento existe uma determinada escassez de profissionais habilitados a atuar naquela área, portanto, a educação a distância tem um papel fundamental neste ponto, pois permite que os poucos profissionais existentes possam direcionar a transmissão de conhecimento para uma quantidade maior de pessoas utilizando determinadas ferramentas, que possibilitam que um emissor transmita a sua mensagem a uma grande quantidade de receptores ao mesmo tempo, pois são poderosos mecanismos de difusão da informação.

Um dos fatores que contribuiu para o desenvolvimento da Educação a Distância foi o surgimento das redes de computadores e consequentemente da Internet, favorecendo o desenvolvimento de sistemas computacionais que viabilizam comunicação à distância e a troca de dados entre usuários geograficamente distantes conectados por seus dispositivos eletrônicos: PC's (*Personal Computers*), notebooks, PDA'S (*Personal Digital Assistants*), ou ainda telefones celulares. Isso vem substituindo gradativamente, o modelo de educação a distância que ainda utiliza o correio tradicional (Litwin, 2001). Esses sistemas são conhecidos como sistemas distribuídos, e os tipos mais utilizados pela EaD são: as videoconferências, os fóruns, os *e-mails*, e os *Chats*.

As ferramentas de *Chat*, assim como o e-mail e o fórum, estão entre os sistemas distribuídos mais populares. No entanto, diferentemente dos últimos, o *Chat* é uma ferramenta síncrona, pois necessita que a comunicação aconteça em tempo real, já que todos os participantes precisam estar *on-line* para que possam trocar mensagens. Essa ferramenta se popularizou muito por meio de *Messengers*, que agregam ainda outras funções. Mas o *Chat* ainda apresenta problemas em relação à falta de coordenação da troca de mensagens entre usuários. A compreensão das mensagens de texto é dificultada quando há uma grande quantidade de mensagens enviadas simultaneamente. Essa quantidade tende a aumentar caso o número de participantes também aumente, principalmente se estes forem bastante ativos no envio de mensagens. A compreensão do contexto das conversações diminui, pois o fluxo de mensagens é intenso demais para ser lido e compreendido, pois não há uma sincronia entre os participantes.

Na tentativa de suprir a falta de coordenação em ferramentas tradicionais de *Chat* foi desenvolvido o sistema *SynchroTalk* (Gomes & Cortês, 2006) que agrega, além da funcionalidade básica da troca de mensagens, três mecanismos de coordenação que garantem a um único participante por vez, a oportunidade de enviar mensagens sem a interferência de outros durante a sua participação. No entanto, esses mecanismos de coordenação da troca de mensagens não impedem que esse *Chat* perca sua coordenação caso falhas aconteçam. Consequentemente a ocorrência de falhas acarretam na perda de coordenação nesse bate-papo.

Os sistemas distribuídos podem apresentar a capacidade de se recuperar após a ocorrência de falhas. Essa capacidade é denominada tolerância a Falhas. A recuperação do sistema pode dar-se de maneira parcial ou total, dependendo da técnica de tolerância a falhas utilizada e da gravidade da falha ocorrida. A tolerância a falhas tem se tornado uma característica cada vez mais desejável em sistemas distribuídos, visto que a interrupção dos serviços pode ocasionar sérios prejuízos, dependendo da área de atuação do sistema.

Este artigo vem demonstrar a programação de técnicas de tolerância a falhas para o *SynchroTalk*. O objetivo destas técnicas é possibilitar uma maior confiabilidade dos mecanismos de coordenação desse *Chat*, de forma que este possa ser utilizado em cursos de educação a distância com maior segurança, devido ao fato de que mesmo diante de falhas, esses mecanismos poderão se recuperar e dar continuidade a seus serviços.

2. TRABALHOS CORRELATOS

As ferramentas de bate-papo são exemplos de sistemas distribuídos muito populares entre os usuários de computadores e da Internet. Apesar de não oferecer o mesmo tipo de comunicação de uma videoconferência, onde é possível que as pessoas obtenham em tempo real a imagem uma das outras com alto grau de qualidade e definição de imagem, o *Chat* apresenta-se como uma solução de menor custo, pois a transmissão de vídeo pela Internet requer uma maior velocidade e qualidade na transmissão de dados através da rede, para possibilitar uma comunicação audiovisual de tempo real. As ferramentas de bate-papo trabalham apenas com transmissão de mensagens de texto, o que não requer os mesmos padrões de velocidade e qualidade necessários ao funcionamento de uma videoconferência, embora também funcionem melhor em redes mais velozes.

Apesar de a Videoconferência ser bastante utilizada no contexto das Universidades Virtuais, as ferramentas de bate-papo tem sua importância garantida, pois se torna de vital importância a realização de encontros periódicos através da Internet. Com a utilização do *Chat*, os aprendizes e os orientadores podem trocar conhecimentos mesmo estando em suas residências ou locais de trabalho de forma a propiciar uma maior flexibilidade, pois a comunicação pode se dar à distância desde que os usuários tenham acesso a Internet.

Mesmo com a possibilidade de utilização do *Chat* no contexto da Educação a Distância, este pode apresentar alguns problemas em alguns cenários (Oeiras & Rocha, 2000). Um deles é quando e os aprendizes acabam se distraindo pelo fato de estarem trocando mensagens em particular com outros aprendizes, não concentrando suas atenções nas mensagens que estão sendo enviadas pelo orientador de ensino-aprendizagem.

Um outro cenário que atrapalha a utilização do *Chat* no contexto da Educação a Distância apresenta-se quando muitos ou todos os participantes do bate-papo desejam se manifestar ao mesmo tempo acerca de um determinado tópico que está sendo discutido. Como consequência, ocorre um envio maciço e desordenado de mensagens em um espaço muito curto de tempo. Essa falta de coordenação complica ainda a tarefa do orientador, visto que é difícil responder simultaneamente a tantos questionamentos, e de dar a atenção devida a cada participante.

Percebe-se que a tradicional ferramenta de *Chat* precisa de adaptações para que possa oferecer maiores contribuições a Educação a Distância. O serviço de troca de mensagens provido pela ferramenta já não é suficiente para comunicar com qualidade aprendizes e orientadores em uma reunião virtual. Esse serviço deve fornecer alternativas para que a troca de mensagens seja mais coordenada. Algumas experiências em cursos semi-presenciais e a distância usando o ambiente TelEduc (Rocha, 2002) mostraram a dificuldade dos usuários em utilizar o bate-papo porque essa ferramenta foi incorporada sem considerar esse novo contexto, o educacional, ao qual iria pertencer, visto que inicialmente o *Chat* não foi pensado para o contexto educativo da EaD, ele tinha sido elaborado apenas para comunicar algumas pessoas (Vronay et al., 1999).

O sistema *Chaos* (Hillery, 1999) propõe uma solução com base na definição de papéis sociais para a coordenação de uma sessão em um bate-papo tradicional. O *Threaded Chat* (Smith et al, 1999) e o *Hiperdiálogo* (Pimentel e Sampaio, 2001) adotam abordagens semelhantes. Ambos tentam reduzir a sobrecarga cognitiva do usuário e facilitar os acompanhamentos através diversos núcleos de conversa gerados em uma sessão de bate-papo, pois cada mensagem gera uma linha de diálogo independente.

O sistema *Mediated Chat* (Rezende et al., 2003) propõe o uso de quatro “técnicas de conversação” para auxiliar na organização de uma conversação em etapas bem definidas: a) contribuição livre, na qual todos podem enviar mensagens a qualquer momento; b) contribuição circular, na qual o usuário deve enviar a sua mensagem quando o botão “enviar” estiver habilitado; c) contribuição única, que permite o envio de uma única mensagem, sendo que o botão “enviar” ficar desabilitado após essa ação; d) contribuição mediada, na qual o usuário precisa solicitar a palavra a um do mediador para que possa enviar as mensagens que deseja.

Todos esses sistemas obtiveram dificuldades de aceitação por parte dos usuários, pois exigiam grande esforço cognitivo. Apesar da preocupação em solucionar o problema da falta de coordenação, não foi encontrado na literatura consultada nenhum exemplo de sistema que tenha se preocupado com tolerância a falhas, que é um aspecto imprescindível para que sistemas usados para essa modalidade sejam capazes de continuar operando mesmo diante da presença de falhas. Do contrário, os processos de ensino-aprendizagem que dependem da ferramenta estarão comprometidos, e terão que contar com a não ocorrência de falhas, o que não é garantido, pois as falhas são praticamente inevitáveis.

3. TOLERÂNCIA A FALHAS EM SISTEMAS DISTRIBUÍDOS

Tanenbaum (2003) define que sistemas distribuídos são coleções de computadores interdependentes que se apresentam ao usuário como um sistema único e consistente. Na definição de Coulouris (2000), sistemas distribuídos consistem em uma coleção de computadores autônomos ligados por uma rede de comunicação.

A expansão dos sistemas distribuídos tem sido favorecida pelo barateamento dos equipamentos de hardware, bem como dos softwares distribuídos, mas um dos fatores que mais impulsionou sua utilização é o avanço em relação às tecnologias de transmissão de dados e dos meios físicos de comunicação. Uma das vantagens da aplicação desse tipo de sistema é a possibilidade de seu crescimento incremental, ou seja, novos computadores e linhas de comunicação podem ser inseridos. Outra vantagem é o desenvolvimento de softwares que devem ser inerentemente distribuídos, tais como: o e-mail, a videoconferência e o *Chat*.

Embora o grau de confiabilidade em sistemas distribuídos tenha aumentado, não é possível garantir que essa confiabilidade seja total, pois não é possível afirmar que falhas não irão acontecer. Apesar de falhas serem praticamente inevitáveis, suas consequências tais como: o colapso do sistema, a perda de dados e a interrupção dos serviços oferecidos, podem ser minimizadas por meio de tolerância a falhas.

Tolerância a falhas é a capacidade de um determinado sistema de se recuperar após a ocorrência de falhas e continuar a prestação dos serviços após sua recuperação. Em muitos casos é possível que o sistema venha a se recuperar totalmente e de maneira imperceptível para o usuário. Atualmente uma quantidade cada vez maior de sistemas tem aplicado técnicas de tolerância a falhas, visto que a interrupção dos serviços pode ocasionar graves consequências para os usuários e para os desenvolvedores. Após o avanço tecnológico, a sociedade tem dependido cada vez mais de sistemas computacionais, entre eles os distribuídos. Quanto maior a dependência, maior será o grau de preocupação e a necessidade de aplicação de tolerância a falhas.

Vários fatores devem ser levados em consideração na programação de estratégias de tolerância a falhas, pois a utilização dessas técnicas possui custos associados, que podem ser financeiros(ou seja, os gastos com equipamentos) ou custos de desempenho. As técnicas de tolerância a falhas podem diminuir a performance das aplicações, apesar de aumentarem a confiabilidade das mesmas. Um domínio da área auxilia os desenvolvedores e os empresários a encontrarem a melhor relação custo-benefício.

Parte das técnicas de tolerância a falhas se baseia na replicação de componentes de hardwares e/ou softwares. Dessa forma, caso algum dos componentes venha a falhar, uma réplica do mesmo poderá continuar a prestar o serviço que antes era fornecido pelo componente que falhou. No entanto, a adoção de replicação envolve a necessidade de manter a consistência de informações entre mesmas os componentes que as originaram e as réplicas. Técnicas de eleição também podem ser utilizadas, visto que caso existam vários componentes capazes de realizar a mesma tarefa, outro componente pode ser eleito para tomar o lugar daquele que falhou.

Ao se trabalhar com grupos de comunicação, todo o grupo deve ser informado quando um novo membro for adicionado ou removido do mesmo. É necessário que o sistema usufrua de uma boa infra-estrutura de comunicação e ainda que seja capaz de detectar quando um de seus membros falhou, de forma a poder atualizar todos os membros que ainda estão ativos, garantindo assim a consistência de informações.

4. A FERRAMENTA *SYNCHROTalk*

4.1 Visão Geral da Ferramenta

O *SynchroTalk* é um *Chat* que está sendo desenvolvido pelo Departamento Acadêmico de Informática do CEFET-MA, por integrantes do Grupo de Pesquisa em Sistemas Distribuídos e Computação Móvel. O objetivo dessa ferramenta é servir de apoio para a Educação a Distância, pois poderá ser utilizado para a realização de encontros periódicos entre participantes de cursos à distância. Assim como as demais ferramentas de *Chat* existentes, O *SynchroTalk* oferece um serviço de conversação por mensagens de texto entre os participantes conectados. No entanto, algumas outras funcionalidades foram adicionadas, entre elas o suporte para mecanismos de coordenação da troca de mensagens. O *SynchroTalk* será um software livre.

4.2 Componentes do Sistema

Os componentes do *SynchroTalk* são basicamente três softwares: o servidor, subdividido em três softwares (conexão, requisição e mensagens); o software cliente e; o software de coordenação. Utiliza-se ainda um

quarto software, responsável pelo monitoramento do sistema. No entanto esse não faz parte da arquitetura original do *SynchroTalk*, sendo utilizado apenas para testes. Trata-se do IC2D, uma tecnologia que foi agregada ao *SynchroTalk*.

O servidor de conexão é responsável pela interligação dos clientes que acessam o *SynchroTalk*, sendo ele ainda responsável por permitir ou não uma conexão e avisar aos clientes de todas as conexões existente no bate-papo, de forma que todos os usuários saibam quantos e quais são os participantes da reunião virtual.

O servidor de requisições é responsável pela entrega de pedidos de acesso do servidor de mensagens para o coordenador da reunião virtual.

O servidor de mensagens é o componente responsável pelo despacho das mensagens de texto para todos os participantes do *SynchroTalk*, incluindo o coordenador.

O software cliente é o componente utilizado pelos aprendizes para se conectar ao *SynchroTalk*. Ele é utilizado pelos mesmos para se comunicar com todos os participantes do *Chat*, incluindo o coordenador da sala.

O software coordenador é o componente utilizado pelos orientadores, pelo qual eles poderão além de ter acesso ilimitado ao servidor de mensagens, criar uma sala virtual para a reunião e coordenar a reunião.

4.3 Mecanismos de Coordenação

Os três mecanismos de coordenação da troca de mensagens utilizadas pelo *SynchroTalk* foram programados com base em algoritmos de exclusão mútua distribuída. Esses algoritmos podem ser aplicados em sistemas distribuídos (Maekawa, 2002) quando se deseja obter o acesso exclusivo a um determinado recurso compartilhado entre dois ou mais computadores, ou seja, a finalidade desses algoritmos é garantir que apenas um computador por vez tenha acesso ao recurso, estabelecendo regras de concorrência. Sendo assim, apenas um participante por vez terá acesso ao servidor de mensagens, que nesse caso é o recurso compartilhado que está sendo concorrido. Sempre que o coordenador deseja usar o servidor de mensagens, seu acesso tem prioridade.

Os algoritmos de exclusão mútua que servem de base para os mecanismos de coordenação são: o centralizado, o distribuído e o *token-ring* (*token* = bastão, *ring* = anel). A programação de cada algoritmo deu origem a um mecanismo de coordenação análogo, pois as propriedades são muito semelhantes. Os algoritmos adotados, apesar de serem úteis, não garantem a exclusão mútua em situações de falhas dentro dos sistemas para os quais foram programados. Por essa razão os mecanismos de coordenação do *SynchroTalk* apresentam essa mesma fragilidade (Agrawal, 2002).

O mecanismo centralizado se caracteriza pela importância do papel do coordenador. Quando um dos participantes deseja utilizar o servidor de mensagens, ele deve enviar ao coordenador uma requisição que é entregue através do servidor de requisições. Uma requisição é um tipo de mensagem especial (contendo o recurso desejado e a hora em que foi efetuado o pedido) que indica que um participante está pedindo para enviar mensagens de texto para o bate-papo. O coordenador mantém uma fila de requisições que são atendidas por ordem de chegada. Ao atender uma requisição, o coordenador identifica se o servidor de mensagens está sendo usado naquele momento. Quando um participante deixa de usar o servidor de mensagens, aquele envia para o coordenador uma mensagem de liberação do recurso. Após a liberação, o coordenador permite que o participante cuja requisição foi analisada, tenha acesso ao servidor de mensagens. É perceptível nesse mecanismo que o coordenador torna-se um ponto crítico de falhas por acabar tornando-se um gargalo. Caso alguma falha ocorra no computador do coordenador da sala de bate-papo, esse mecanismo acaba tornando-se ineficiente, visto que o mesmo depende do coordenador para ter acesso ao servidor de mensagens. Caso o participante que detém o acesso ao recurso venha a falhar, ele não terá condições de liberá-lo.

O mecanismo distribuído não necessita de um coordenador, pois o acesso ao servidor de mensagens depende de todos os participantes envolvidos no bate-papo. Neste mecanismo um participante que deseja utilizar o servidor de mensagens, deve enviar uma requisição a todos os demais, visando descobrir se algum deles está ocupando o recurso desejado. Cada participante mantém uma fila de requisições que são atendidas por ordem de chegada. Quando um participante atende uma requisição, seu software cliente pode reagir de três formas. (1) Caso esteja usando o servidor de mensagens: não responde a requisição até terminar de usá-lo. (2) Caso não esteja usando o servidor de mensagens e também não deseja acessá-lo: responde a requisição

indicando que não está acessando o recurso. (3) Caso não esteja usando o servidor de mensagens, mas deseja acessá-lo: compara a hora da requisição que enviou com as das requisições que recebeu. Aquele que enviou a requisição com menor tempo de chegada no servidor de requisições, tem o acesso ao recurso desejado.

Consequentemente cada participante pode ser um ponto crítico de falhas, visto que todos dependem de todos para ter acesso ao servidor de mensagens. Sendo assim, uma falha em qualquer computador conectado ao *SynchroTalk*, impede que os participantes ativos tenham suas requisições respondidas. Não há como saber se um participante está demorando a responder, ou se este falhou.

O mecanismo *token-ring* consiste em organizar de maneira virtual todos os participantes do *Chat* em forma de anel. Pelo anel de participantes deverá circular um *token*. Este simboliza a chave de acesso ao servidor de mensagens, portanto só poderá acessá-lo quem estiver com o *token*. Quando o *token* chega a qualquer participante, ele é avisado da sua presença, e é questionado se deseja ou não acessar o servidor de mensagens. Caso responda sim, ele ganha o direito de enviar suas mensagens de texto e quando terminar ele deverá passar o *token* ao próximo participante do anel. Caso a resposta seja não, o *token* é simplesmente passado adiante. O coordenador pode tomar para si o *token* a qualquer hora, pois tem privilégios dentro do bate-papo.

O problema desse mecanismo é a perda do *token*, que poderá acontecer caso o computador do participante que o detém venha a falhar, pois não terá como repassar o *token* adiante.

Além das possibilidades de falhas mencionadas para cada mecanismo, deve-se adicionar a possibilidade de falhas no servidor. Caso este falhe existem algumas conseqüências: (1) Novos clientes não poderão se conectar devido às falhas no servidor de conexões e tão pouco os já existentes poderão ser avisados de futuras desconexões; (2) Requisições de acesso ao servidor de mensagens não poderão ser entregues ao coordenador por conta das falhas no servidor de requisições e; (3) Mensagens de texto não poderão mais ser enviadas.

4.4. Tecnologias Utilizadas

Para o desenvolvimento do *SynchroTalk*, foi utilizado um conjunto de tecnologias inteiramente gratuitas, pois o objetivo era não gerar custos com licenças de software para os usuários finais, e não criar dependências financeiras em relação a nenhuma empresa de tecnologia. Isso possibilitará que *SynchroTalk* possa ser utilizado por instituição que não tem condições de arcar com custos tecnológicos, pois será um software livre. Entretanto o fato de serem tecnologias gratuitas não implica em baixa qualidade. Trata-se de tecnologias respeitadas e que já tem estudos relevantes no meio acadêmico, mas que também são largamente utilizados no desenvolvimento de softwares para empresas, pois estas que estão cada vez mais interessadas em diminuir custos com o pagamento de licenças de tecnologias proprietárias. As tecnologias adotadas possuem licença de código aberto, conhecidas como *open source* LGPL, ou seja, é possível analisar como foram programadas e ainda alterá-las de acordo com as necessidades de cada sistema, possibilitando melhorá-las.

A linguagem de programação utilizada para a construção do sistema foi a *Java* (Sun, 2007). Sua escolha deu-se em função de alguns fatores, entre eles está a grande variedade de recursos disponíveis pela mesma para o desenvolvimento de softwares desde os mais simples até os mais complexos, e sua capacidade de aplicação em uma grande variedade de dispositivos que vão desde notebooks, computadores pessoais e servidores até dispositivos móveis como PDA's e telefones celulares. Outra motivação é a característica dessa linguagem de gerar softwares que funcionam em qualquer sistema operacional, seja ele Linux, Windows, ou qualquer outro, sem alterações no código que gerou a aplicação ou sem necessidade gerar um novo tipo de software específico para cada sistema operacional, desde que possua uma *JVM* (Maquina Virtual Java), que é responsável pela interpretação da linguagem. *Java* é conhecida pelos seus mecanismos de segurança, pois a *JVM* não permite a execução de programas nocivos ao computador.

Para possibilitar que os participantes do *SynchroTalk* pudessem se comunicar uns com os outros através de uma conexão pela Internet com o servidor, foi utilizado o *middleware*¹ *ProActive* (ObjectWeb, 2007), que é

¹ O *middleware* é uma camada de software que é responsável pela comunicação entre o sistema distribuído e o sistema operacional. O *middleware* é usado quando dois ou mais sistemas precisam se comunicar.

compatível com a linguagem *Java* e foi construído utilizando a mesma linguagem. Entre os vários recursos desse *middleware* está a capacidade de gerenciar grupos de comunicação, esta capacidade foi usada no gerenciamento dos participantes do *SynchroTalk*.

As tecnologias citadas no desenvolvimento do *SynchroTalk* são aplicáveis quando se deseja adotar técnicas de tolerância a falhas, em especial o *ProActive*, que permitem detectar falhas dentro de um grupo de comunicações, desta forma o *SynchroTalk* usa esse recurso para saber quando um participante do bate-papo já não é mais capaz de se comunicar com o servidor. Além disso, o software de monitoramento denominado *IC2D* é fornecido pelo *ProActive*. O *IC2D* (já mostrado na figura 1) permite ainda injetar falhas no *Chat*

Os ambientes de programação (também conhecidos como IDE's) utilizados para se trabalhar com a linguagem *Java* foram o *Eclipse* e o *NetBeans*. Essas ferramentas permitem maior produtividade, visto que elas auxiliam na correção de erros de sintaxe e permitem criar interfaces gráficas de usuários por meio de desenho de componentes e ajudam no gerenciamento de projetos de software.

5. APLICAÇÃO DE TOLERÂNCIA A FALHAS NO SYNCHROTALK

Para tentar diminuir as consequências das falhas as quais estão sujeitos os mecanismos de coordenação do sistema *SynchroTalk* foram aplicadas técnicas de replicação, eleição, e detecção de falhas em grupos de comunicação. Para cada mecanismo de coordenação foram aplicadas uma ou mais técnicas, dependendo da necessidade que cada mecanismo apresenta. Foram programadas ainda técnicas para caso de falhas no servidor.

Para o caso do mecanismo de centralizado foi programado um mecanismo eletivo, que permite que após a falha do coordenador, eleger um novo coordenador. Sendo assim sempre que o coordenador vigente da sala de bate-papo não puder mais se comunicar com os demais, um novo coordenador será eleito entre aqueles participantes que ainda estão ativos. Ainda para esse mecanismo utiliza-se uma técnica de detecção de falhas em grupos de comunicação, o que permite que o coordenador vigente saiba quais participantes do grupo ainda estão ativos, de forma que caso seja detectada alguma falha no participante que tinha o acesso ao servidor, o coordenador atenderá automaticamente o próximo participante que pleiteia enviar suas mensagens de texto.

Para o caso do mecanismo distribuído a detecção de falhas no grupo de participantes do *Chat*, também foi aplicada. Quando um participante envia uma requisição a todos os demais, este deixa de esperar pela resposta de qualquer participante que esteja inativo. Sendo assim ele poderá sair do estado de bloqueio em que se encontrava e poderá continuar a concorrer pelo acesso ao servidor de mensagens. Caso o participante que tenha falhado seja aquele que possuía o acesso do servidor, será enviada uma resposta de liberação do recurso para todos os demais participantes, fazendo com que a concorrência seja retomada a partir de então.

Para o caso do mecanismo *token-ring* foi adotada a técnica de replicação, combinada com a técnica de detecção de falhas em grupos de comunicação. Sempre que for detectada uma falha em algum participante, será verificado se a falha ocorreu no participante que detinha o *token*. Caso isso seja confirmado, todos os uma réplica do *token* perdido é gerada. Graças a réplica, é possível continuar o mecanismo de coordenação, pois a réplica garante o acesso ao servidor de igual forma ao *token* perdido.

Para o caso de falhas nos componentes de software do servidor, por hora optou-se apenas por uma mudança dinâmica da arquitetura do sistema em combinação com mecanismos de detecção de falhas. Sempre que um software cliente conectado ao *Chat* percebe que não consegue mais se comunicar com o servidor, ele abre uma conexão direta com todos os demais participantes, adotando um comportamento *P2P* (*Peer-to-Peer*), e passa a enviar suas mensagens de texto e requisições diretamente sem intermédio do servidor, que já não é mais capaz de atender os clientes.

A mudança arquitetural para *P2P* é simplificada pelo *ProActive*, porém trata-se de uma solução meramente temporária, pois novas técnicas estão sendo estudadas e aplicadas, entre elas a de redundância de hardware e software, de forma a permitir que outra máquina de igual poder de processamento possa trabalhar como servidor, contendo réplicas dos componentes de software do servidor para que os mesmo serviços sejam oferecidos.

6. RESULTADOS E DISCUSSÕES

Os testes realizados consistiram em simulações de cenários de falhas para testar a eficiência das técnicas de tolerância a falhas que foram aplicadas para o sistema *SynchroTalk*. Esses testes foram realizados em uma rede local dentro do CEFET-MA.

Para realizar a simulação foram injetadas falhas na comunicação entre os participantes usando a ferramenta IC2D e também desligando algumas máquinas de forma a fazer o sistema reagir diante das falhas que estavam sendo injetadas. Ainda não houve preocupação em quantificar o tempo de recuperação, pois a atual fase de pesquisa ainda está se concentrando na aplicação dessas e de novas técnicas. Apenas após o período de programação dessas técnicas serão iniciados os testes de desempenho e de comparação entre as técnicas adotadas e entre os mecanismos nos quais foram aplicadas.

A aplicação das técnicas de tolerância a falhas no *SynchroTalk* resultou em mecanismos de coordenação um pouco mais confiáveis, visto que eles são capazes de suportar alguns tipos de falhas. No entanto, ainda são necessárias técnicas que repliquem os dados existentes no servidor, para que outro servidor possa responder pelo primeiro, caso este falhe. Entretanto, essa não é uma tarefa trivial, pois envolve uma grande complexidade em relação a como realizar a replicação sem comprometer o desempenho das atividades do servidor. Outro motivo pela não confiabilidade total nos resultados obtidos, é que eles são provenientes de testes realizados em rede local, portanto ainda é prematuro dizer que o desempenho dessas técnicas é ideal, visto que a velocidade de transmissão de dados em redes locais é teoricamente sempre maior do que na Internet. Futuramente pretende-se utilizar um emulador de rede para mensurar os valores de desempenho, visto que o emulador possibilitará atingir-se um comportamento mais próximo do real. Pesquisas futuras deverão revelar que emulador se adequará melhor aos objetivos dos testes.

A aplicação das técnicas de tolerância a falhas trouxe alguns custos de desempenho para aplicação, visto que quanto mais máquinas falharam, maior foi o trabalho de processamento para sinalizar aos membros do grupo de participantes a ocorrência de falhas, e ainda para atualizar o grupo de comunicação.

Percebe-se ainda que há um problema de escalabilidade quando o *SynchroTalk* muda para a arquitetura *P2P*. Apesar de o *middleware ProActive* garantir esse modo de comunicação graças a seus recursos de gerenciamento de grupos, não há como novos usuário se conectarem ao bate-papo, pois não há como conhecer o endereço de rede de algum participante que já esteja conectado para informar os endereços dos demais participantes conectados.

Contudo, é visível a manutenção de comunicação entre aqueles que já estavam conectados, preservando ao menos que professores e alunos que já estavam em contato possam permanecer assim, até que um novo servidor possa retomar o controle das conexões e do envio de mensagens.

A volta do servidor é importante ainda para garantir que todos os usuários recebam as mensagens de texto na mesma ordem, visto que as mesmas são entregues pela ordem de chegada no servidor. No modo *P2P* não há como garantir essa ordem (o que é um outro problema gerado pela mudança arquitetural), pois não há um ponto de convergência para uma posterior distribuição, o que explica o fato de não adotar-se o modo *P2P* desde o início. Uma problemática a ser resolvida será programar um mecanismo que permita a detecção da volta do servidor e o redirecionamento do envio de mensagens para este, bem como informar ao servidor quais usuário já estão conectados, para que a entrega de mensagens seja feita aos devidos destinatários.

Apesar das técnicas descritas, outras ainda serão testadas. Somente após a aplicação de todas as técnicas pesquisadas e desejadas, e da realização dos testes utilizando não apenas emuladores, mas a própria Internet, é que será possível avaliar quais os custos e os benefícios que as técnicas de tolerância a falhas trouxeram para o *SynchroTalk*.

As tecnologias utilizadas se mostraram satisfatórias. Ainda falta explorar muitos recursos oferecidos tanto pela linguagem *Java*, quanto do *middleware ProActive*. No entanto, a pesquisa tem buscado sempre observar o lançamento de novas versões dessas tecnologias, de forma a manter o sistema mais atualizado possível. Novas tecnologias também estão sendo analisadas para fazer calcular os custos e os benefícios que trariam uma possível mudança de tecnologias, sempre priorizando a adoção de tecnologias livres e de boa qualidade.

7. CONSIDERAÇÕES FINAIS

O desenvolvimento do *SynchroTalk* tem mostrado que sistemas distribuídos tradicionais podem ser adaptados para que sejam melhor utilizados no contexto da educação a distância. É possível concluir que existe a possibilidade de aplicar mecanismos de coordenação não apenas para mensagens de texto em *Chats*, mas que outros sistemas distribuídos tais como a videoconferência poderiam ser também coordenados. No caso da videoconferência é possível pensar que o tráfego de voz e vídeo poderia ser coordenado de forma semelhante ao *SynchroTalk*, permitindo que apenas a voz a imagem de uma pessoa por vez seja transmitida.

As pesquisas têm permitido observar que os mecanismos de coordenação só são úteis caso eles possam ter um relativo grau de confiabilidade. Portanto, é imprescindível que o *SynchroTalk* possa oferecer mecanismos de coordenação que sejam capazes de suportar a maior variedade de falhas possível, visto que num contexto real de utilização dos mesmos na Internet, não há como evitar que falhas na comunicação aconteçam, e por isso é necessário que o sistema esteja preparado para elas, com técnicas adequadas para cada tipo de falha.

É possível observar que cada mecanismo de coordenação possui características específicas. Portanto, antes de decidir qual mecanismo é o mais adequado para o contexto da educação a distância, é necessário saber qual deles é o mais confiável em caso de falhas. De posse dessas informações será possível determinar se a escolha do melhor mecanismo de coordenação deverá levar em consideração apenas fatores adaptação ou de confiabilidade, ou seja, é necessário saber se um mecanismo deve ser escolhido pelo grau de eficiência com o qual coordena o servidor de mensagens, ou pela capacidade de recuperação diante de falhas.

Do ponto de vista pedagógico o sistema de bate-papo proposto vem a contribuir com a Educação à Distância, pois além de permitir formas normais de comunicação à distância, o *SynchroTalk* possui mecanismos de coordenação que permitem uma melhor organização dos encontros virtuais, uma vez que seus recursos de controle impedem que várias pessoas enviem mensagens ao mesmo tempo. Nesse contexto, debates, aulas, discussões, entre outros tipos de atividades cognitivas poderão ser desenvolvidas com maior grau de segurança caso falte a colaboração no *Chat*. Sendo assim, os orientadores terão formas de melhorar a organização do espaço virtual. Contemplando o ponto de vista tecnológico, a ferramenta oferece a possibilidade de funcionamento em qualquer sistema operacional e possibilitou aos envolvidos o aprendizado de tecnologias.

8. REFERÊNCIAS

- AGRAWAL, D, **An efficient and fault-tolerant solution for distributed mutual exclusion**. ACM Transactions on Computer Systems, vol. 9, no. 1, Feb. 2002, p. 1-20.
- COULOURIS, GEORGE & DOLLIMORE, JEAN & KINDBERG, Tim, **Distributed Systems: Concepts and Design**, Adson Wesley, 3ªed, 2000.
- GOMES, BERTO DE T. P; CORTÊS, OMAR ANDRÉS. C. **Sincronização em uma Aplicação Distribuída para Educação a Distância Utilizando Recursos do Middleware ProActive**. Em: VII Workshop em Sistemas Computacionais de Alto Desempenho e Simpósio Internacional de Arquitetura de Computadores e Computação de Alta Performance, Ouro Preto-MG, 2006. Centro de Convenções UFOP. Anais, pág. 179-182
- HILLERY, P. **On-line chats sessions! Chaos or ...?** Em: TCC '99 Papers, 1999. Capturado em 11 de abril. 2006. On-line. Disponível na Internet em <http://leahi.kcc.hawaii.edu/org/tcon99/papers/hillery.html>.
- LITWIN, E. **Educação à Distância: Temas para o Debate de uma Nova Agenda Educativa**, Artmed, Porto Alegre-RS, 2001.
- MAEKAWA, M., **A sqrt(n) algorithm for mutual exclusion in decentralized systems**, ACM Transactions on Computer Systems, vol 3, no. 4, May 2002, pp. 145-159.
- OBJECTWEB, **web site do middleware Proactive**: <http://www.objectweb.org/ProActive>, ultimo acesso 25 de Agosto de 2007.
- OEIRAS, J. Y. Y.; ROCHA, H. V. **Uma Modalidade de Comunicação Mediada por Computador e suas Várias interfaces**. Em: Workshop sobre Fatores Humanos em Sistemas Computacionais, 2000, Gramado. Anais. Porto Alegre: Instituto de Informática da UFRGS, págs. 151-160.

PIMENTEL, M. G.; SAMPAIO, F. F. **Hiperdiálogo uma ferramenta de bate-papo para diminuir a perda de contexto**. Em: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE, 2001, Novembro. 21-23, Vitória - ES. Anais: págs. 255-266.

ROCHA, H. V. **O ambiente TelEduc para educação a distância baseada na web: Princípios, funcionalidades e perspectivas de desenvolvimento**, em MORAES, M. C. (Org.) Educação a distância: Fundamentos e práticas. Campinas, SP: UNICAMP/NIED, 2002, cap.11, págs.197- 212.

REZENDE, J. L.; FUCKS, H.; LUCENA, C. J. P. **Aplicando o protocolo social através de mecanismos de coordenação embutidos em uma ferramenta de bate-papo**. Em: Simpósio Brasileiro de Informática na Educação, 2003, Rio de Janeiro. Anais. Porto Alegre: Instituto de Informática da UFRGS, p. 55-64.

SMITH, M.; CADIZ, J. J.; BURKHALTER, B. **Conversation trees and threaded chats**. SIGCHI Bulletin, Minneapolis, v. 31, n. 3, p. 21-23, Jul., 1999. Capturado em 9 de Dezembro de 2006. On-line. Disponível na Internet em: <http://www.acm.org/sigchi/bulletin.3/morales.pdf>

SUN, **web site da tecnologia Java**: Disponível em: <http://java.sun.com/products/jdk/rmi/>, último acesso em 25 de Agosto de 2007.

TANENBAUM, ANDREW S, **Sistemas Operacionais Modernos**, São Paulo: Prentice Hall, 2003.

VRONAY, D.; SMITH, M.; DRUCKER, S. **Streaming media interfaces for chat**. Redmond: Microsoft Research, 1999. (Technical Report. MSR-TR-99-75). Capturado em 9 de dezembro de 2002. On-line. Disponível na Internet. <http://research.microsoft.com/scg/papers/dilemmachi2000.pdf>