

DIFERENCIAÇÃO DE SERVIÇOS EM AMBIENTES VIRTUAIS COLABORATIVOS EM GRANDE ESCALA NO ENSINO A DISTÂNCIA

João Paulo de Brito Gonçalves - jpaulo@ifes.edu.br – Instituto Federal do Espírito Santo – Campus Cachoeiro de Itapemirim
Paulo César Salgado Vidal - vidal@de9.ime.eb.br - Instituto Militar de Engenharia, Departamento de Engenharia e Sistemas

RESUMO. *Ambientes virtuais colaborativos (AVCs) baseados na Internet apresentam problemas de desempenho potenciais, já que a Internet não provê nenhuma garantia de largura de banda ou atraso. Este artigo apresenta os estágios de desenvolvimento de um modelo de simulação para um protocolo de comunicação para AVCs. Para aumentar a confiabilidade deste protocolo, ele foi combinado com técnicas de qualidade de serviço e mais tarde modelado em um simulador, para avaliação de desempenho por meio de simulação.*

Palavras-chave: *Ambientes Virtuais Colaborativos, Qualidade, Simulação*

ABSTRACT. *Internet-based Collaborative Virtual Environments(CVEs) present potential performance problems, as the Internet does not provide any guarantees of bandwidth or delay. This paper presents the stages of development of a simulation model for a protocol of communication for CVEs. To increase the reliability of this protocol, it was combined with techniques of quality of service and later shaped in a simulator, for performance evaluation through simulation.*

Keywords: *Collaborative Virtual Environments, Quality, simulation.*

Submetido em 05 de dezembro de 2015.
Aceito para publicação em 14 de maio de 2016.

POLÍTICA DE ACESSO LIVRE

Esta revista oferece acesso livre imediato ao seu conteúdo, seguindo o princípio de que disponibilizar gratuitamente o conhecimento científico ao público proporciona sua democratização.

1. INTRODUÇÃO

Os Ambientes Virtuais Colaborativos (AVCs) são sistemas de realidade virtual que possibilitam o compartilhamento do mundo virtual por usuários geograficamente dispersos. Dentre os diversos tipos de aplicações para as quais estes ambientes podem ser usados pode-se citar: treinamento, telemedicina, arquitetura, visualização científica e principalmente Educação a Distância onde podem trabalhar em conjunto com um AVA tradicional, como a plataforma Moodle (MOODLE, 2014) e até em dispositivos móveis como podemos ver em Voss et al. (2013) e Marçal et al. (2005).

Os AVCs que possuem um grande número de usuários ou uma extensão geográfica elevada são conhecidos como Ambientes Virtuais Colaborativos em grande escala. A maioria destes ambientes utiliza a Internet como meio de comunicação e troca de mensagens devido ao grande número de usuários que a utilizam em todo mundo, mas atingir interação multiusuário com estes ambientes na Internet, onde a largura de banda é geralmente escassa, sempre foi considerado um grande desafio.

Dessa forma, como a capacidade de rede é um recurso limitado, a plataforma de um AVC deve ser cuidadosamente projetada para alocar os recursos disponíveis para os vários tipos de informação que os diversos usuários devem trocar, de forma a evitar que a rede se torne um gargalo para a aplicação e o aluno perca o senso de interatividade no AVA. Exemplos de plataformas deste tipo são a SIMNET (GARVEY et al., 1998), a NPSNET (MACEDONIA et al., 1994), e a SPLINE (ANDERSON et al., 1993).

Na fase de inicialização, ou seja, quando o usuário está tentando se conectar ao sistema, o mundo virtual junto com seu conteúdo deve ser enviado para o novo usuário. Uma vez que o usuário se une à aplicação, atualizações de eventos são geradas quando um usuário cria/remove um objeto, transfere sua propriedade, deixa o sistema ou se move dentro do mundo virtual. Devido à diversidade de mensagens que transitam durante uma sessão de um AVC em grande escala, é necessário que existam garantias específicas em relação à largura de banda, atraso e perda de pacotes para que seja possível manter a consistência e a interatividade entre as aplicações, como por exemplo em web conferências. Estas garantias podem ser obtidas por meio de uma arquitetura de Qualidade de Serviço (*Quality of Service* - QoS). Uma arquitetura de qualidade de serviço tem o objetivo de prover às aplicações os recursos computacionais que elas necessitam para executarem adequadamente, principalmente em períodos de congestionamento da rede.

Em AVCs, é desejável tratar pacotes de tipos de mensagens diferentes de formas diferentes na rede. Por exemplo, atualizações de estado relacionadas a novos usuários se unindo ao mundo virtual são mais importantes e urgentes que atualizações de eventos e podem ser relacionadas a um serviço de maior prioridade, que as conceda maiores garantias com relação aos recursos que necessitam. A diferenciação de serviços é, portanto, uma técnica de qualidade de serviço adequada para Ambientes Virtuais Colaborativos em Grande Escala, pois permite o tratamento de tráfegos mais prioritários de forma diferenciada na rede.

Este trabalho propõe o emprego da técnica de diferenciação de serviços em uma plataforma para Ambientes Virtuais Colaborativos em grande escala, apresentando as etapas de desenvolvimento de um modelo de simulação da plataforma SPLINE para transmissão de mensagens de comunicação entre os usuários remotos. Pretende-se demonstrar que a utilização da técnica de diferenciação de

serviços aplicada a uma plataforma de um AVC, não só aumenta o desempenho desta como também ainda permite a entrega de mensagens críticas com confiabilidade e pouco atraso já que será dado um tratamento diferenciado a estas mensagens, garantindo assim que elas obtenham maiores garantias de entrega e atraso em relação às outras mensagens da aplicação.

1.1 SPLINE

A plataforma SPLINE (*Scalable Platform for Large Interactive Networked Environment*) foi desenvolvida no laboratório *Mitsubishi Electric Research Labs* (MERL) (WATERS et al.,1997). Na plataforma, o mundo virtual é modelado por uma base de dados que contém informações sobre este mundo, chamada de Modelo de Mundo. Este modelo está parcialmente distribuído entre os processos SPLINE. As aplicações interagem umas com as outras por meio da modificação desta base de dados e da observação das modificações feitas por outros usuários.

Na plataforma SPLINE, o mundo é dividido em áreas chamadas *locales*, que não possuem tamanho nem forma fixas e podem ser processadas separadamente. Cada *locale* está associado a um grupo multicast e possui um sistema de coordenadas próprio. Cada participante possui uma cópia parcial da base de dados de acordo com o seu *locale*. Servidores de *locale* são responsáveis por manter um registro do estado de todos os objetos em um determinado *locale*. Dessa forma, um processo SPLINE que se interessa por um *locale* pode receber a informação atualizada sem ter que contatar cada processo pertencente a ele.

Para a comunicação entre processos, é usado o protocolo ISTP (*Interactive Sharing Transfer Protocol*). O principal objetivo do ISTP é encaminhar informações sobre os objetos em um Modelo de Mundo compartilhado em um Ambiente Virtual Colaborativo. Tal objetivo inclui características para gerenciar os vários objetos presentes em um Mundo Virtual. Os vários objetos que a SPLINE deve suportar carregam características bem diferentes, indo de objetos muito extensos e estáticos, como uma imagem de fundo, até objetos em tempo real, como áudio ou vídeo. Para conseguir atender tal gama de características, o protocolo ISTP é composto de 5(cinco) subprotocolos:

- subprotocolo de conexão ponto-a-ponto: usado para estabilizar e manter uma conexão TCP (*Transmission Control Protocol*) entre dois processos ISTP;
- subprotocolo de transmissão de estado do objeto: usado para comunicar o estado de objetos de um processo ISTP para outro. Tais atualizações podem ser enviadas por uma comunicação ponto-a-ponto ou por uma comunicação multicast via o protocolo UDP (*User Datagram Protocol*)
- subprotocolo de áudio em cadeia: usado para enviar áudio na simulação;
- subprotocolo de comunicação baseado em *Locales*: é o núcleo do protocolo ISTP e suporta o compartilhamento de informações sobre objetos no modelo de mundo e o correto gerenciamento do particionamento do mundo virtual;
- subprotocolo de comunicação baseado no conteúdo: suporta comunicação de informação sobre *beacons* usando o modelo de servidor central. *Beacons* podem ser considerados objetos do Mundo Virtual que exercem algum tipo de função específica no ambiente.

Os últimos dois subprotocolos são construídos sobre os outros três, pois os primeiros criam uma infraestrutura de comunicação e o compartilhamento necessários para fornecer o suporte para os dois últimos protocolos. Uma característica chave do ISTP é que ele é totalmente distribuído, ou seja, um processo ISTP tanto pode ser o cliente ou o servidor para outros processos.

Dentre as mensagens da plataforma SPLINE, trocadas em uma sessão, três são de particular interesse para nosso experimento. Elas são:

- *Object State* – contém a descrição de um ou mais objetos que um processo possui ou as mudanças ocorridas em seu estado desde o último envio de uma mensagem *Object State* por aquele processo.
- *Object State Summary* – mensagens usadas para que um processo servidor possa comunicar a um processo cliente informações sobre todos os objetos que este processo conhece ou para que um processo cliente possa comunicar a um servidor todos os objetos que ele possui.
- *Locale Com Status* – são mensagens usadas para regular a comunicação baseada em *Locales*.

A plataforma SPLINE foi escolhida para este trabalho devido ao seu vasto uso na implementação de ambientes virtuais no meio acadêmico (ANDERSON, 1993).

1.2 Qualidade de Serviço

A Qualidade de Serviço pode ser definida como uma linha de pesquisa que busca o desenvolvimento de técnicas para prover classes de serviços diferenciados e níveis de prioridade para fluxos de dados. Seu objetivo é garantir que os tráfegos de aplicações críticas como os AVCs, por exemplo, tenham um desempenho aceitável na rede, mesmo em períodos de congestionamento. Em situações onde a largura de banda é limitada e os AVCs e outras aplicações multimídia como web-conferência por exemplo, concorrem por recursos escassos, a Qualidade de Serviço se torna uma ferramenta fundamental para garantir que todas as aplicações possam coexistir e funcionar com níveis de desempenho adequados.

As técnicas de Qualidade de Serviço têm como objetivo comum acelerar a entrega de pacotes de aplicações críticas, enquanto compartilham os recursos da rede com aplicações não-críticas. Elas também dão aos gerentes de rede controle sobre as aplicações de rede, aumento na relação custo-eficiência de redes WAN (*Wide Area Network*) e permitem uma diferenciação de serviços permitindo que desenvolvedores de aplicações façam um balanceamento dos níveis de prioridade dos serviços para a satisfação dos usuários com eficiente utilização de acesso para minimizar a latência da rede. Eficientemente priorizando uma variedade de tipos de tráfego e requerimentos para tráfego de aplicações críticas e sensíveis ao tempo como os AVCs, tais técnicas trazem uma série de benefícios como melhor controle sobre recursos, uso dos recursos da rede de forma mais eficiente, coexistência entre aplicações críticas e as fundações para uma rede totalmente integrada no futuro.

Apesar da necessidade do uso de técnicas de qualidade de serviço em seus módulos de comunicação, a grande maioria das plataformas para AVCs em grande escala não possuem estas características incorporadas aos seus modelos. O trabalho de Eraslan et al. (2003) apresenta uma implementação de um AVC com um módulo de diferenciação de serviços acoplado à sua plataforma VESIR-6 (*Virtual Environment*

Multi-user Interaction over IPv6). Nesta plataforma, as mensagens que transportam comandos de controle de mudança de estado têm uma prioridade maior no encaminhamento do que as mensagens de atualização de eventos.

O trabalho de Greenhalgh et al. (1999) apresenta uma arquitetura para replicação de dados com qualidade de serviço em AVCs. Nesta arquitetura a qualidade de serviço é alcançada por meio da predição de carga e adaptação às variações de tráfego. A negociação do tipo de serviço permite ao servidor controlar os recursos alocados por cliente.

2. DESENVOLVIMENTO

Esta seção apresenta as etapas de desenvolvimento da plataforma SPLINE no simulador de redes NS-2 (*Network Simulator*) (FALL, 1996; VARADHAN, 1996). O objetivo da simulação é avaliar o comportamento da plataforma utilizando aplicações com diferentes requisitos de qualidade de serviço. A simulação compreende as operações críticas, como a troca de grupos, entrada e saída do mundo, e as atualizações de estado.

Para simular a plataforma SPLINE com a utilização de técnicas de qualidade de serviço, foi definido um subconjunto de funções e mensagens do protocolo ISTP, usadas pela plataforma. No modelo simplificado do protocolo ISTP proposto, apenas três dentre os cinco subprotocolos foram desenvolvidos no simulador. Os subprotocolos do ISTP implementados são os seguintes:

- Subprotocolo de conexão ponto-a-ponto;
- Subprotocolo de Transmissão de Estado de Objeto;
- Subprotocolo de Comunicação baseado em Locales.

Com a implementação dos três subprotocolos citados acima, foi possível gerar grande parte do tráfego de mensagens necessário para gerar resultados na simulação que possam ser posteriormente analisados. As mensagens usadas pela aplicação que foram implementadas são: *Object State*, *Object State Summary* e *Locale Com Status*.

O protocolo ISTP foi implementado de duas formas no simulador NS: sem o uso de técnicas de qualidade de serviço e com o uso.

Na implementação do protocolo ISTP sem o uso de técnicas de qualidade de serviço, as mensagens de controle (*Locale Com Status*) são enviadas em conexões ponto-a-ponto TCP e as mensagens de atualização (*Object State*, *Object State Summary*) são enviadas via protocolo UDP.

Por outro lado, na implementação usando a diferenciação de serviços todas as mensagens são enviadas por meio do protocolo UDP e são usadas classes de serviços diferentes para prover garantias específicas para cada tipo de mensagem. No caso das mensagens de controle, elas recebem uma prioridade especial em relação à outra classe de mensagens, permitindo com isso, que elas tenham uma maior largura de banda e um menor atraso. Com isso, foi eliminado o atraso referente ao estabelecimento da conexão TCP entre os processos e com garantia na entrega das mensagens de controle. A sequência da transmissão das mensagens implementada no simulador está descrita abaixo:

- após o início da conexão, uma mensagem *Locale Com Status* requisitando que a comunicação seja iniciada deve ser enviada pelo processo requisitante para o servidor de *Locales*;

- quando uma mensagem *Locale Com Status* for recebida pelo processo requisitante como resposta à sua mensagem anterior, deve ser aberta uma conexão para o endereço multicast especificado na mensagem e que foi informado pelo servidor de *Locales* e deve-se começar a receber ou enviar informação.
- a partir disso, mensagens *Object State* e *Object State Summary* serão enviadas pelo processo servidor, como todas as informações que ele possui sobre os objetos existentes naquele *Locale*;
- periodicamente (uma vez a cada 30-100 milissegundos), um processo envia uma ou mais mensagens *Object State* descrevendo todos os objetos que ele possui naquele *Locale* que mudaram desde a última vez que ele enviou mensagens. Tais mensagens são enviadas via multicast. Da mesma forma, este processo deverá processar as mensagens *Object State* recebidas de outros processos.

As mensagens têm um tamanho definido como um parâmetro que será uma estimativa do tamanho em bytes de cada mensagem, respeitando a definição do tamanho de cada campo apresentada em Waters et al. (1997). Entretanto, para fins de simplificação, os campos de cada mensagem não foram implementados. O tamanho em bytes das mensagens simuladas pode ser visualizado na Tabela 1.

Os objetivos das simulações executadas foram avaliar a quantidade de mensagens perdidas nos dois modelos, tanto a perda total de mensagens quanto a perda das mensagens de controle na simulação e a quantidade média de mensagens necessária para a abertura de uma sessão no mundo virtual. A partir da mudança nos valores dos parâmetros usados, foram simulados vários cenários, e seus resultados posteriormente analisados.

Tabela 1 - Tamanho das mensagens em bytes

Mensagem	Tamanho(bytes)
Locale Com Status	126
Object State	100
Object State Summary	1000

A topologia criada é composta de uma árvore *multicast* cheia com 16 nós, sendo que o servidor de *Locales* é ligado à um roteador interno a rede e esse enlace é o gargalo da rede com uma largura de banda de 1Mbps. Os demais enlaces possuem uma largura de banda de 10 Mbps e atraso de 2 milissegundos. Tal topologia foi escolhida porque a árvore binária com enlace de gargalo representa uma situação crítica, onde existem enlaces que concentram a maioria do tráfego em uma rede. Estes enlaces estão localizados próximos da fonte de um grupo *multicast*, o qual é um cenário passível de existência na Internet. Além disso, o enlace com gargalo também é considerado um ponto crítico com relação à perda de mensagens, pois a perda de uma mensagem de controle prejudica a entrada de um nó na simulação do ambiente virtual.

O modelo de tráfego utilizado para simular a aplicação é composto de mensagens de controle (*Locale com Status*, *Object State Summary*) que são enviadas

no início de cada interação de cada usuário com a simulação. Estas mensagens são trocadas entre o usuário que deseja participar da simulação ou que deseja sair e o servidor de *Locales* que irá aprovar ou rejeitar sua requisição.

Após isso, são geradas mensagens de atualização com 100 bytes de tamanho a cada 50 milissegundos em média, gerando um tráfego constante no período que essas mensagens de atualização (*Object State*) estão sendo enviadas. Usando a semente aleatória do NS-2, foi criada uma dinâmica de entrada e saída dos nós do mundo virtual. Dessa forma, quando um usuário deseja se conectar ao mundo virtual, ele deve enviar uma mensagem *Locale Com Status* com tamanho de 126 bytes para o servidor de *Locales*. Quando a conexão é aceita (não foi incluído na simulação o caso no qual o servidor de *Locales* rejeita o pedido do usuário), o servidor de *Locales* também envia uma mensagem de resposta com um tamanho médio de 126 bytes. Após a conexão ser aberta, o servidor envia mensagens *Object State Summary* com um tamanho aproximado de 1000 bytes para atualizar o nó que está entrando na simulação em relação a todos os eventos acontecidos no ambiente virtual até então e dos quais ele não participou.

Após estes passos iniciais, o tráfego se torna constante, com mensagens de atualização sendo enviadas a cada 50 milissegundos, em média. Quando um usuário deseja se desconectar do ambiente virtual, ele também deve enviar uma mensagem *Locale Com Status* para o fechamento de sua sessão. O servidor de *Locales* deverá enviar uma mensagem de resposta autorizando a desconexão e atualizando as bases de dados dos outros usuários para que eles retirem a representação virtual do usuário desconectado de seus Modelos de Mundo. Com isso, durante a participação do usuário no Mundo Virtual existe um tráfego constante de 2000B/s gerado por cada nó participante.

Foram criadas duas classes de tráfego: uma com maior prioridade, que corresponde à taxa de transmissão dos pacotes das mensagens de controle e outra classe correspondente à taxa de transmissão dos pacotes das mensagens de atualização. Com isso, o fluxo correspondente às mensagens de controle teria um tratamento prioritário em relação aos pacotes correspondentes ao fluxo das mensagens de atualização. Dessa forma, a taxa para que os pacotes sejam considerados dentro do perfil era de até 1000B/s. Como as mensagens de atualização possuem uma taxa de transmissão muito maior que as mensagens de controle, sua taxa de transmissão facilmente ultrapassa as mensagens de controle, e com isso estes pacotes são marcados como fora do perfil, recebendo um tratamento menos prioritário em relação aos pacotes dentro do perfil.

Para o estudo de vários cenários de congestionamento diferentes, foi usado o modelo de erro do NS-2. A aplicação do modelo variou de uma taxa de 1% a 5% de perda de pacotes. No gargalo que liga o servidor de *Locales* ao resto da topologia, foi mantida uma taxa de erro constante de 5%. Estes valores representam uma taxa de perda de pacotes considerada média.

Na primeira simulação não foi usado nenhum modelo de taxa de erros, por isso a situação de congestionamento gerada na simulação foi decorrente do próprio envio de dados das aplicações em enlaces com capacidade menor que as taxas de transmissão da aplicação.

3. CONCLUSÃO

O principal parâmetro analisado nas simulações foi a perda de pacotes, tanto de mensagens de atualização quanto de mensagens de controle. Foi analisada a perda de pacotes média coletada a cada 30 segundos da simulação. Os scripts foram executados com valores aleatórios diversas vezes, e a taxa média de perda foi calculada baseada nos resultados destes scripts. O intervalo de confiança foi calculado para o nível de confiança de 95%, usando-se uma amostra de trinta repetições e usando-se uma distribuição Normal. Os intervalos de confiança usados para cada intervalo da simulação, referentes a cada 30 segundos de simulação podem ser vistos na Tabela 2.

Tabela 2– intervalos de confiança para a simulação de perda de pacotes

Intervalo da Simulação	Intervalo de Confiança
0.5	338,9 + 13,8172
1.0	934, 2 + 10,22525
1.5	1223,9 + 10,78454
2.0	1608,9 + 42,90923
2.5	2386,9 + 22,71724
3.0	3749,8 + 21,22799
3.5	5945,6 + 29,04184
4.0	6911,5 + 20,62931
4.5	7450,6 + 21,07746
5.0	8480 + 18,51561

A Figura 1 mostra o comportamento da plataforma simulada sem o uso de diferenciação de serviços e com o uso. Ela mapeia a perda de pacotes no decorrer dos 5 minutos da simulação.

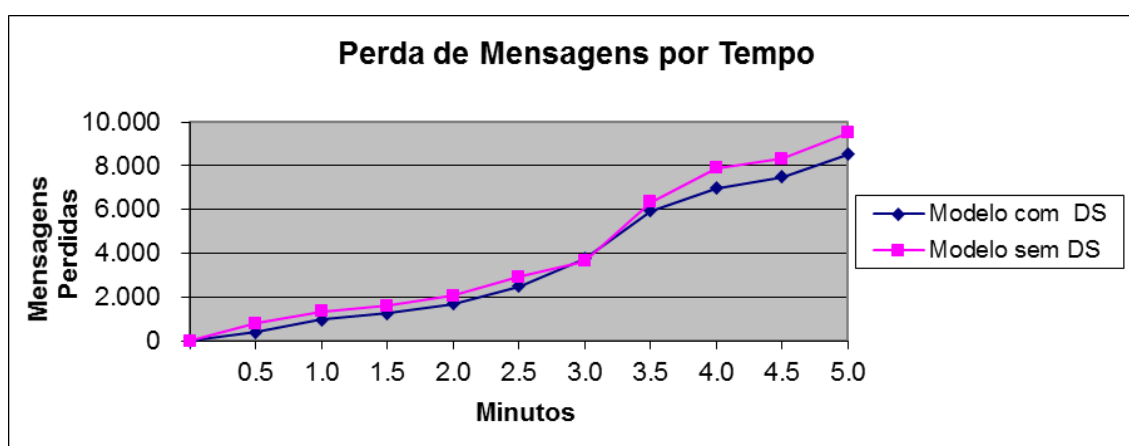


Figura 1 – Número de mensagens perdidas com e sem o uso da diferenciação de serviços. Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Foram enviadas aproximadamente 80.420 mensagens, somadas as mensagens de atualização, controle e próprias à comunicação multicast. É possível observar que o número de pacotes perdidos tem um crescimento mais intenso a partir do 2,5 minuto de simulação, onde um número maior de fontes de tráfego transmite simultaneamente seu tráfego constante de mensagens de atualização. Pode ser observado também, que para quase todo o tempo da simulação, existe uma tendência

de que a quantidade média de perda de pacotes para a plataforma sem o uso da diferenciação de serviços foi maior, com 12,5% de aumento em relação ao modelo que usava esta técnica. A alta perda de pacotes pode ser explicada por uma baixa capacidade nos buffers das filas de mensagens, que em uma situação de congestionamento acabam por descartar um número excessivo de mensagens.

Já na Figura 2 é mostrado o comportamento das duas plataformas em relação à perda das mensagens de controle. Nas simulações deste tipo, houve uma transmissão média de 215 mensagens de controle. Para este caso, as simulações realizadas sugeriram que o uso da diferenciação de serviços também se mostrou eficaz, com uma tendência de perda 5,67% menor que o modelo sem o uso.

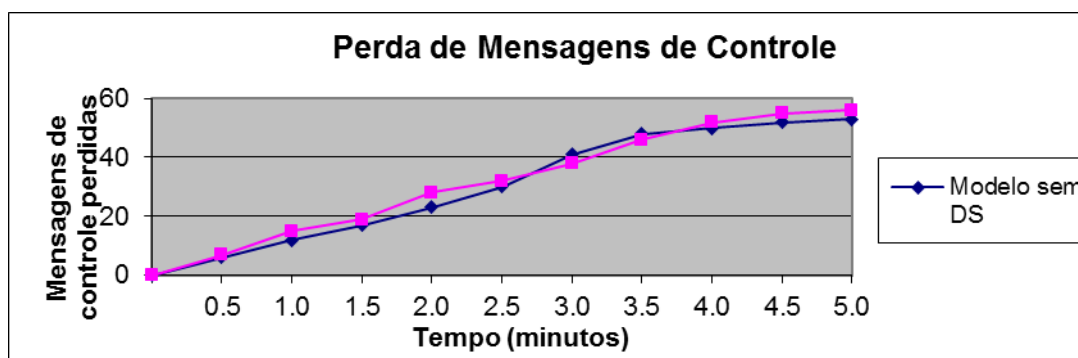


Figura 2 – Variação da perda de mensagens de controle com e sem o uso da diferenciação de serviços. Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Aproximadamente entre o 3º e 4º minuto de simulação, o modelo sem diferenciação de serviços apresenta uma taxa de perda menor que o modelo que usa a técnica, mas no resto do período da simulação fica clara a vantagem do modelo que adota a técnica em relação ao outro modelo.

Nas simulações deste tipo, houve uma transmissão média de 215 mensagens de controle. Para este caso, as simulações realizadas sugeriram que o uso da diferenciação de serviços também se mostrou eficaz, com uma tendência de taxa de perda 5,67% menor que o modelo sem o uso. Aproximadamente entre o 3º e 4º minuto de simulação, o modelo sem diferenciação de serviços apresenta uma taxa de perda menor que o modelo que usa a técnica, mas no resto do período da simulação fica clara a vantagem do modelo que adota a técnica em relação ao outro modelo.

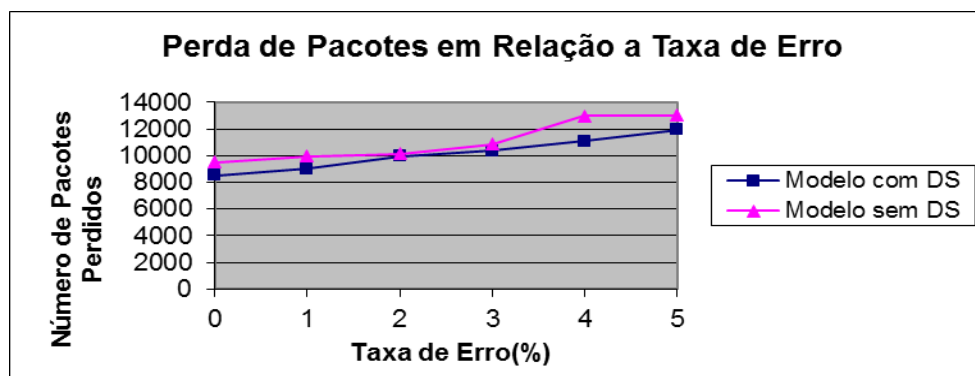


Figura 3 – Variação da perda de mensagens de controle com e sem o uso da diferenciação de serviços. Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Após os resultados coletados sem o uso da diferenciação de serviços, foi aplicada uma taxa média de perdas, variando de 1% a 5% para que fosse possível analisar o comportamento da plataforma em um cenário de congestionamento. As simulações foram executadas diversas vezes para cada taxa de perda e depois foi tirada uma média dos resultados obtidos. A Figura 3 mostra o comportamento da plataforma para os dois modelos com a aplicação da taxa média de perdas. Pode ser observado que a taxa média de perda de pacotes, calculada com base em todos os resultados gerados pelos scripts sugere um aumento de 9,13% no modelo sem o uso de diferenciação de serviços.

Em relação às mensagens de controle, também acontece uma tendência à uma melhora significativa com a aplicação da diferenciação de serviços. Na figura 4, possível observar que quanto maior vai se tornando a taxa de erro dos enlaces, maior se torna a diferença na taxa de perda entre os dois modelos. Neste caso, a taxa média de perda de pacotes sugere cerca de 10,2% de aumento nas simulações que não usam a diferenciação de serviços.

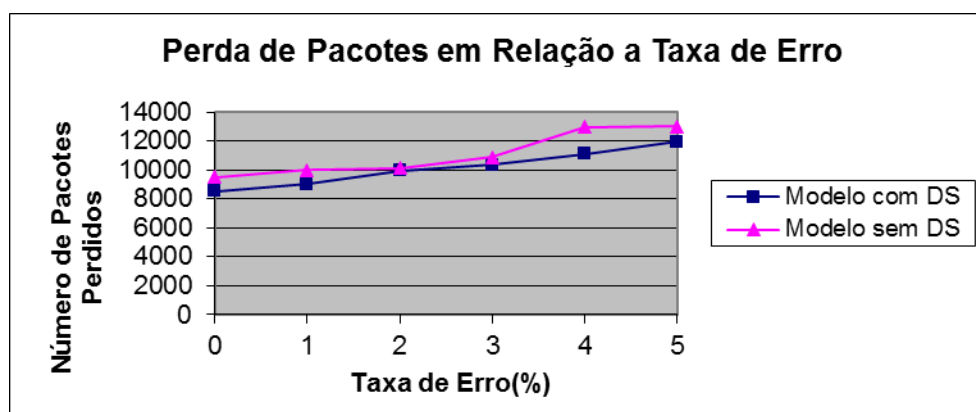


Figura 4 – Variação da perda de mensagens de controle com diferentes taxas de erro nos dois modelos. Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Outro aspecto analisado pelas simulações foi o número médio de mensagens necessárias para se abrir uma conexão com o mundo virtual nos dois modelos.

Por conta de no modelo sem o uso da diferenciação de serviços o protocolo TCP é utilizado, acontece uma retransmissão das mensagens perdidas enviadas tanto pelo nó quanto pelo servidor. Já no modelo que usa apenas o protocolo UDP, não acontecem essas retransmissões, e desta forma o nó emissor deve recomeçar o processo de comunicação com o servidor normalmente.

A Figura 5 apresenta os dados coletados nas simulações para diferentes taxas de erros nos dois modelos. Foi possível inferir deste gráfico que, inicialmente, a abordagem utilizando diferenciação de serviços leva uma pequena vantagem no número de mensagens necessárias que o modelo que usa o protocolo TCP, devido ao tratamento prioritário que as mensagens de controle recebem nas filas, que fazem com que exista uma perda menor deste tipo de mensagem.

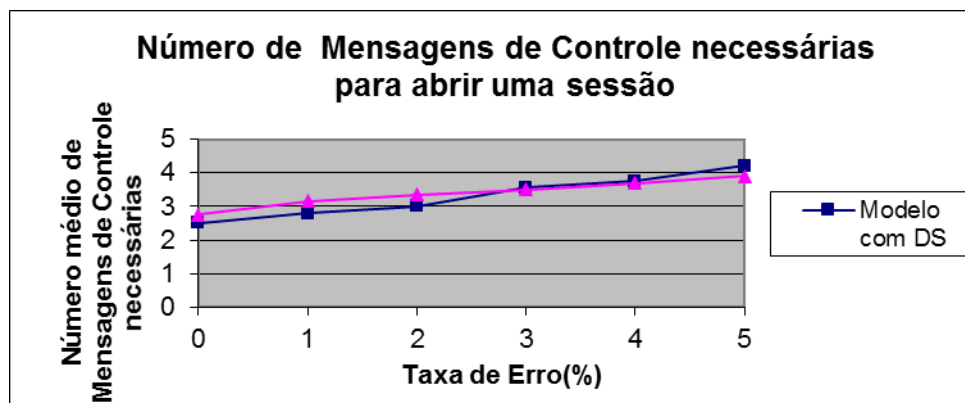


Figura 5 – Número médio de mensagens de controle necessárias para abertura de uma sessão na simulação nos dois modelos. Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Entretanto, é possível observar que à medida que a taxa de erro cresce na simulação, o modelo que usa o TCP começa a ter uma pequena vantagem em relação ao modelo com diferenciação de serviços. Isso pode ser explicado pelo fato de que quando uma mensagem de controle é perdida com o TCP, acontece uma retransmissão apenas daquela mensagem. Já em relação ao outro modelo, quando acontece uma perda de mensagem, seja de requisição de abertura de sessão ou de resposta do servidor de *Locales*, a única alternativa é o processo requisitante começar novamente a comunicação, enviando uma mensagem para o servidor de *Locales*. Com isso, sempre que uma mensagem de controle é perdida neste modelo, são necessárias ao final, quatro mensagens de controle para que a conexão seja estabelecida enquanto que no outro modelo, seriam necessárias três mensagens quando da perda de uma mensagem. Dessa forma, ainda que a diferenciação de serviços garanta uma perda menor de mensagens de controle, quando uma delas é perdida, acontece o envio de mais duas mensagens o que ocasiona o aumento no número médio de mensagens necessárias.

4. CONSIDERAÇÕES FINAIS

Este trabalho teve por objetivo analisar a utilização de uma técnica de Qualidade de Serviço para otimizar o desempenho na transmissão de mensagens de um Ambiente Virtual Colaborativo em Grande Escala.

Por meio da análise dos resultados coletados nas simulações é possível constatar que, ainda que as simulações tenham sugerido vantagens em relação ao uso da diferenciação de serviços nos modelos analisados, a utilização de protocolos de comunicação confiáveis com mecanismos de retransmissão para envio de mensagens críticas à aplicação não pode ser de todo descartada no projeto da arquitetura de comunicação de um ambiente virtual colaborativo em grande escala.

Com o crescimento do uso da Educação a Distância como forma democrática de acesso ao ensino de qualidade pela população, deve-se pensar em abordagens adequadas para o funcionamento dos ambientes virtuais colaborativos na Internet, para que sejam ferramentas complementares aos Ambientes Virtuais de Aprendizagem e propiciem uma aprendizagem verdadeiramente significativa aos educandos.

REFERÊNCIAS

- ANDERSON, D. B., et al., **Diamond Park and Spline: A Social Virtual Reality System with 3D Animation, Spoken Interaction, and Runtime Modifiability**, Tech. Report 96-02, MERL, Cambridge, Mass, 1996.
- BARRUS J. W.; WATERS, R.C.; ANDERSON, D. B. Locales: Supporting Large Multi- User Virtual Environments. **IEEE Computer Graphics and Applications**. p.50-57, 1996.
- ERASLAN, M.; GEORGANAS, M. D.; GALLARDO, J.R.; MAKAKRIS, D., **A Scalable Network Architecture for Distributed Virtual Environments with Dynamic QoS over IPv6**, The Eighth IEEE Symposium on Computers and Communications, 2003.
- FALL, K.; VARADHAN, K., **NS Notes and Documentation, Technical Report**, The VINT Project, 1996.
- GARVEY, R. E.; MONDAY, P. **SIMNET SIMulator NETwork**, BBN Technical Note. 1998.
- GREENHALGH C., BENFORD, S. **A Multicast Network Architecture for Large Scale Collaborative Virtual Environments, Multimedia Applications, Services and Techniques – ECMAS'T'97**, 1997.
- GREENHALGH C., BENFORD, S., e REYNARD, G. **A QoS architecture for collaborative virtual environments** . In Proceedings of the ACM Multimedia, Orlando, p.121-130, 1999.
- MACEDONIA, M. et al. NPSNET: a Network Software Architecture for Large Scale Virtual Environments. **PRESENCE: Teleoperators and Virtual Environments**, v. 3, n. 4, p.265-287, 1994.
- MARÇAL, E.; ANDRADE, R.; RIOS, R. Aprendizagem utilizando Dispositivos Móveis com Sistemas de Realidade Virtual. **RENOTE - Revista Novas Tecnologias na Educação**, CINTED-UFRGS . v. 3, n.1, Maio, 2005. Porto Alegre, 2005.
- MOODLE.ORG (s/d). Sítio Web de apoio à plataforma Moodle. Disponível em: < <https://moodle.org/>>. Acesso em: 06 out. 2014.
- VOSS, G. B., NUNES, F. B., HERPICH F. & MEDINA, R. D. **Ambientes Virtuais de Aprendizagem e Ambientes Imersivos: um estudo de caso utilizando tecnologias de computação móvel**. XXIV Simpósio Brasileiro de Informática na Educação - SBIE 2013, Campinas-SP, 2013.
- WATERS, R.C.; ANDERSON, D. B.; SCHWENKE, D. L. **Design of the Interactive Sharing Transfer Protocol**. In: IEEE WORKSHOPS ON ENABLING TECHNOLOGIES: Infrastructure for Collaborative Enterprises, v.6, 1997.