Rede Overlay de Suporte a Ambientes Virtuais Colaborativos em Redes Veiculares

Rodolfo Ipolito Meneguette[1]; Regina Borges de Araujo[2]
[1]*Universidade Estadual de Campinas – UNICamp*
[2]*Universidade Federal de São Carlos – UFScar*
ripolito@ic.unicamp.br[1]; regina@dc.unicamp.br[2]

Abstract

Collaborative virtual environments (CVEs) over Vehicular ad hoc networks (VANETs) offer the possibility of user interaction to simulations (from training to games) while inside a moving car. Interactive simulations, such as CVEs, pose tough challenges when ran over networks with high mobility nodes. This paper presents an overlay network to support collaborative virtual environments over VANETs using a modified version of the pure Gnutella protocol overlay.

1. Introdução

Os ambientes virtuais colaborativos (AVCs) são ambientes sintéticos 3D compartilhados por múltiplos usuários localizados remotamente, com um objetivo comum, que acessam o ambiente por meio de avatares, representações gráficas do usuário. Os AVCs vêm ganhando campo em dispositivos móveis, como dispositivos de mão, celular, entre outros, através de jogos e simulações, visando o entretenimento e o conforto do usuário participante. Esses ambientes virtuais requerem uma baixa latência e uma alta escalabilidade para que um grande número de participantes possa interagir em um ambiente consistente e seus usuários não percam a sensação de imersão. Com isso, exige-se uma preocupação maior na estratégia de conexão entre os participantes. Esse cuidado aumenta quando esses ambientes são executados em veículos em movimento, já que as características singulares de uma rede ad hoc veicular (VANETs), como alta mobilidade dos nós, alta velocidade de transmissão e uma topologia altamente dinâmica, requerem uma estrutura complexa que possa superar essas limitações. Essas características nos levam aos desafios da comunicação entre os participantes e à disseminação do resultado das ações de seus *avatares* no AVC e/ou mudança de estado de um objeto para todos os participantes. A rede de suporte à comunicação deve suportar a entrada de um número crescente de participantes no AVC sem que isso afete a sensação de presença desses usuários.

De modo a superar os desafios de escalabilidade e inconsistência espaço/temporal na visualização de aplicações de AVCs em redes VANETs, este trabalho apresenta o projeto e avaliação de uma rede *overlay*, mais precisamente uma rede *peer-to-peer* (p2p), que visa a prover conectividade entre os usuários participantes de um mesmo AVC, além da atualização de estado de seus *avatares* e de objetos no AVC para todos os participantes, garantindo que essas informações sejam entregues no menor tempo possível. Um dos objetivos de uma rede *overlay* é abstrair a complexidade das camadas inferiores de rede para a aplicação. Redes *peer-to-peer overlay* possuem, em sua essência, características similares às de uma rede veicular, tais como topologia dinâmica e auto-organização [5], [7]. Diferentemente de outros modelos de rede, as redes *peer-to-peer* não degradam com a sobrecarga de mensagens como em um modelo cliente/servidor [2] [4], tampouco com o tempo de migração de um servidor para outro em um modelo de multi-servidor [10], o que pode propiciar escalabilidade para AVCs emVanets.

2. Desenvolvimento de um Suporte *Overlay* para Ambientes Virtuais Colaborativos sobre as Redes Veiculares

Esse trabalho tem como objetivo a implementação e avaliação de uma rede *overlay* de suporte a ambientes virtuais colaborativos (AVCs) em Redes Ad Hoc Veiculares, com objetivo de auxiliar na comunicação entre os usuários participantes, diminuindo o atraso na entrega das mensagens para a aplicação, e abstraindo a complexidade da rede veicular para a aplicação.

Tomamos como foco a simulação de treinamento de equipes de preparação e resposta a emergências por meio de dispositivos em veículos enquanto estes estão sendo

dirigidos a um local de acidente. Assim, o treinamento no ambiente virtual que reflete o ambiente real, enquanto em trânsito, pode ajudar na tomada de decisão das equipes de comando quando estas chegam ao local, reduzindo perdas de vida e de patrimônio.

Assumimos que os ambientes virtuais colaborativos considerados são divididos em hexágonos iguais, como visto na Figura 1, gerando um conjunto de célula, também chamada de zona [1], onde usuários participantes de uma mesma célula formam um grupo de participantes com o mesmo interesse. Essa divisão visa diminuir a quantidade de mensagens que irá trafegar na rede, pois o usuário participante somente enviará informações para aqueles participantes que pertencem à sua célula. Consideramos também que o usuário já possui um identificador único podendo ser, por exemplo, as coordenadas dadas pelo dispositivo de GPS *(Global Positioning System)*. Essa identificação é utilizada para identificar a representação geométrica do usuário participante (*avatar*) tanto em relação ao ambiente virtual quanto à rede veicular. Na solução sendo proposta, todos os participantes são considerados confiáveis, isto é, não têm a intenção de prejudicar a simulação, e todo o ambiente virtual e seus componentes já serão de posse do usuário, isto é, o veículo já possui tanto o hardware como o software para executar o ambiente virtual.

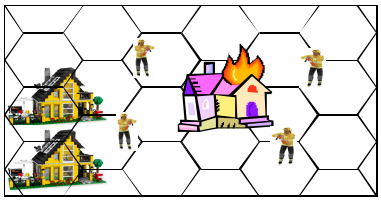

Figura 1 - Divisão do ambiente virtual em hexágonos iguais

2.1 Descrição do funcionamento da Rede *Overlay* Proposta (Gnutella modificada)

Para se iniciar a rede é necessária a entrada de um usuário participante em uma determinada simulação ou jogo que possui um identificador único, como, por exemplo, o identificador de um treinamento. O usuário entra no processo de busca de usuários para verificar se tem algum participante ativo. Para isso, é enviada uma mensagem ping_avc, através de um broadcast, responsável por qualificar o protocolo gnutella, ou seja, verificará a versão do protocolo gnutella. Caso o usuário tenha uma zona favorita, poderá buscar essa zona, habilitando a identificação da zona.

Cada participante ativo, ao receber o ping_avc, verifica a distância entre ele e a origem para analisar se irá aceitar essa primeira conexão ou não, e envia um pong_avc como resposta. Ao aceitar a conexão, o participante passa a ser o coordenador para aquele usuário e adiciona esse novo participante à sua tabela de vizinhos com o identificador do usuário, tempo de envio de mensagem e à área de interesse. A qualidade da conexão é dada pelo tempo de resposta das informações. Caso esse tempo comece a extrapolar um intervalo de aceitação t o nó do usuário começa a buscar novas conexões.

Após esse processo de seleção do cenário da simulação e busca de usuários, é mostrado para o participante quais são as zonas ativas e o número de participantes que estão atuando naquelas zonas. Ao selecionar uma zona, uma solicitação de entrada é enviada na simulação. Essa solicitação é mapeada em uma query_avc que é enviada para o coordenador correspondente, ou seja, para o nó a que o usuário solicitante está conectado. Ao receber uma query_avc, esse nó verifica se está participando daquela zona ou se tem conhecimento de algum nó que está participando e responde com uma queryhit_avc (com informações sobre a zona), preenche a sua tabela de zona, que armazena as informações sobre os nós que estão participando na zona. O nó origem, ao receber a queryhit_avc, envia uma mensagem HTTP para os outros participantes da mesma zona que iniciarão a transferência das informações das atualizações do ambiente (coordenadas de localização, direção, velocidade e uma flag que informará o estado do usuário). Caso o usuário queira deixar a simulação ele modifica essa flag e deixa o ambiente virtual.

Se um nó desejar mudar de uma zona para outra ele verifica novamente em sua tabela de vizinhos se algum nó já está participando e envia uma query_avc direcionada para ele. Caso não exista nenhum participante, ele envia essa mensagem para o seu coordenador e, ao receber uma querhit_avc nova ele apenas fecha a conexão anterior.

A verificação dos nós ativos e inativos da rede é feita através do envio das mensagens ping_avc e pong_avc, da mesma forma que é feita no protocolo gnutella padrão. Porém, o ping_avc só é enviado para os nós pertencentes à mesma área de interesse (zona). Se o tempo do pong_avc estiver com um atraso muito grande e não receber nenhuma confirmação de nova conexão, uma informação de alerta é mostrada para o usuário informando que ele está saindo da cobertura da rede. Caso esse tempo ultrapasse um intervalo t, esse usuário passa a não ter mais nenhuma colaboração na rede até se

reconectar à rede.

Se o usuário deseja sair da aplicação, ele modifica a *flag* de estado e envia essa informação para somente os participantes que pertence a sua zona, avisando sua saída. Posteriormente fecha a conexão e realiza a saída do usuário. A desconexão abrupta, isto é, quando há uma falha de hardware ou de conexão, não é tratada neste trabalho – nesse caso seria necessária a existência de apoio de infra-estrutura externa para que se possa armazenar o estado do ambiente.

2.2. Projeto de Rede *Overlay* de suporte a AVCs em redes Veiculares

Para o desenvolvimento desse suporte utilizamos o protocolo Gnutella [3], por ser um protocolo *open source* [6] e por ser amplamente utilizada [1] por pesquisadores em todo mundo. Foram adicionados campos em algumas mensagens bem como modificado alguns campos para diminuir a quantidade de mensagens trafegada na rede bem como dar maior dinamismo na comunicação entre o ambiente virtual e a rede *overlay* como podemos ver pela Figura 2 (a). as principais mensagens de uma gnutella padrão, o ping, pong, query e queryhit, e do seu lado as mensagens modificadas o ping_avc, pong_avc, query_avc e queryhit_avc como podemos ver na Figura 2 (b)

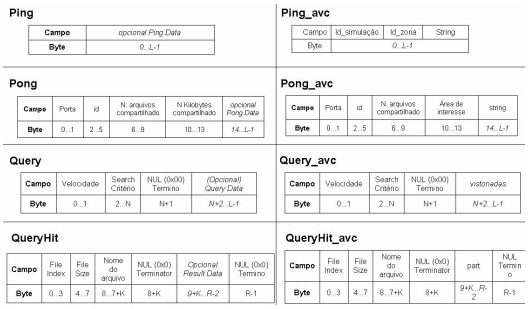

Figura 2- (a) mensagens originais da rede Gnutella; (b) mensagens modificadas baseadas na rede Gnutella

Na mensagem ping realizado um acréscimo no campo opcional da ping data, introduzindo a identificação da simulação, da zona e a string de conexão, que irá filtrar a quantidade de pong recebida como também fará a primeira conexão na rede, não necessitando a elaboração de outra mensagem. Já em sua mensagem de resposta o pong foi adicionado no campo opcional da pong a string de conexão que trará a confirmação ou não da conexão, e modificamos o campo numeram de kilobytes compartilhado para área de interesse, pois identificará a zona atual do usuário como também quais serão as prováveis zonas de interesse desse nó. Na mensagem de busca query acrescentamos no campo opcional da query data o campo vistoriado, pois com esse acréscimo diminuirá a quantidade de query enviadas na rede conseqüentemente não sobrecarregando a rede. Na queryhit Foi realizado um acréscimo no campo opcional da queryhit data com o campo participante, pois indicara quais são os outros nós participantes da zona desejada. O roteamento da gnutella também foi modificado, pois a mensagem de resposta não faz o mesmo o caminho de volta realizado pela mensagem original, mas pode ir por outros caminhos mais curtos para chegar.

3. Experimentos de Simulação

A solução proposta neste trabalho, chamada de **gnutella_avc** é comparada a uma rede gnutella padrão. O objetivo principal dessa comparação é avaliar o desempenho da gnutela_avc e a gnutella padrão segundo métricas descritas nas seções abaixo, utilizando o simulador jist/swans++[9].

Foram utilizadas a latência da rede e a inconsistência espaço-temporal relativa [8] como métricas para a avaliação da rede Overlay proposta e implementada.

A inconsistência espaço-temporal relativa utilizamos para avaliar o impacto sofrido na visualização do ambiente virtual. Essa métrica foi concebida por pesquisadores em [8], fornecendo o tempo e a distancia que o objeto está inconsistência no mundo virtual.

Os parâmetros para a aplicação e rede estão listados na Tabela 1, juntamente com o tempo do envio das mensagens de atualização pela rede.

Tabela 1 - Parâmetros de Simulação da aplicação e rede

Aplicação	
Espaço mínimo entre os participantes	0,1m
Velocidade media dos participantes	0,3 m/s
Diferença entre o clock do ambiente virtual de cada participante	0,01 s
Diferença do posicionamento do avatar	0,3 m
Velocidade máxima do avatar	2 m/s
Participantes	15 entidades
Rede	
Modelo de Mobilidade	Random waypoint
Número de nós na rede veicular	150 veículos
Número de nós na rede overlay	5-10-15 participantes
Velocidade maxima	20 Km/h
Velocidade minima	1 Km/h
Duração da simulação	5 minutos

3.1. Resultados

Foram realizadas comparações desse trabalho com um protocolo gnutella padrão que verificara se as alterações realizadas superam o padrão gnutella. Sendo realizados 30 testes para cada cenário, com diferentes números de entidades participantes do ambiente virtual tomando uma variação de 5,10, 15 entidades, tendo em cada simulação uma duração de 300 segundos (5 minutos) para avaliar a média da latência gerada por essas mensagens de atualização

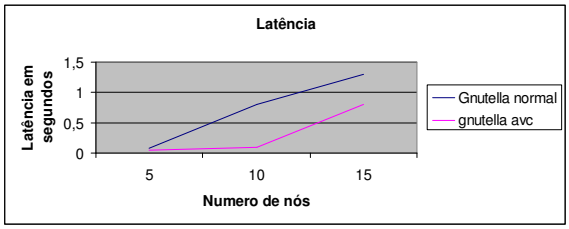

Figura 3 - gráfico da latência gerado pela simulação

Na Figura 3 podemos observar que a gnutella AVC gerou uma latência menor que a gnutella normal devido a liberdade de roteamento e as mudanças realizadas nas mensagem para a realização da troca de zona no ambiente virtual. O pico com 10 usuários se deu devido a posição física dos participantes no momento de enviar e receber as informações.

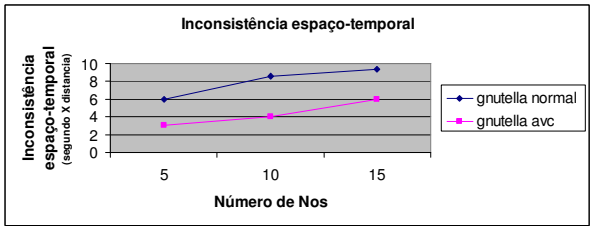

Figura 4 - Gráfico da inconsistência espaço-temporal da visualização do ambiente virtual

A Figura 4 mostrou que a inconsistência gerada pela gnutela normal foi maior que a gnutella AVC, pois a inconsistência está diretamente ligada à latência gerada pela rede. Podemos observar essa influência analisando a simulação com 10 entidades, na qual foi gerado um pico de latência, conseqüentemente afetando a sua consistência.

4. Conclusão

Podemos verificar que a gnutella AVC teve um melhor desempenho do que a gnutella normal, devido a sua forma de gerenciamento da zona, das modificações realizadas para dar maior dinamismo ao ambiente bem como o fim da forma padrão de rotemanto da gnutella normal. Como trabalhos futuros a realização de novas simulações para obter novos resultados desempenho do *overlay*. Desenvolvimento de um protótipo para ver o resultado real dessa *overlay*, bem como utilizar técnicas de *handoff* pasra extrair um melhor desempenho de diferentes redes sem que caia a conexão.

5. Referências

[1] A. Boukerche, A. Zarrad, R.B. Araujo, "*A Novel Optimized Caching Technique for Mobile Gnutella Based Network to Support Large-Scale Collaborative Virtual Environment Simulation* Symposium", 41st Annual Simulation Symposium (anss-41 2008); Ottawa, Abril 2008, pp. 289 - 297

[2] G. Singh, L. Serra, W. Png, and H. Ng. "*BrickNet: A Software Toolkit for Network-Based Virtual Worlds*". Presence: Teleoperators and Virtual Environments, vol. 3, n.1, 1994, pp. 19-34.

[3] Gnutella. Diponivel em:
http://www.gnutella.com acessado em 2010

[4] I. S. Pandzic, K. Tolga, L. Elwin, N. Thalmann, and D. Thalmann. "*A Flexible Architecture for Virtual Humans in Networked Collaborative Virtual Environments*". Proceedings Eurographics, Hungary, 1997, pp. 143-152

[5] K. C. Lee, Seung-H. Lee, R. Cheung, U. Lee, e M. Gerla, "*First Experience with CarTorrent in a Real Vehicular Ad Hoc Network Testbed*", 2007 Mobile Networking for Vehicular Environments, Anchorage, 2007, pp. 109-114.

[6] M. R. Macedonia, M. J. Zyda, D. R. Pratt P. T Barham, S. Zeswitz "*NPSNET: a Network Software Architecture for Large Scale Virtual Environments*", In Presence, vol. 3, n. 4, Fall, 1994, pp. 256-287

[7] S. Ahmed e S. S. Kanhere, "*VANETCODE: a Network Coding to Enhance Cooperative Downloading in Vehicular Ad-Hoc Networks*", Proc. ACM International Conference on Communications and Mobile Computing, Vancouver, British Columbia, 2006, pp. 527 - 532

[8] S. Zhou, W. Cai, Bu-S. Lee, e S. J. Turner "*Time-Space Consistency in Large-Scale Distributed Virtual Environments*", ACM Transactions on Modeling and Computer Simulation, vol. 14, n. 1, 2004, pp. 31-47.

[9] SWANS++ - Extensions to the Scalable Wireless Ad-hoc Network Simulator
http://www.aqualab.cs.northwestern.edu/projects/swans++/

[10] T. K. Das, G. Singh, A. Mitchell, P. S. Kumar, e K. McGee, "*NetEffect: A Network Architecture for Large- scale Multi-user Virtual Worlds*". Proceedings ACM Virtual Reality software and tecchnology, USA, 1997, pp. 157–163.