

# Modelo de Simulação de Tráfego de um Jogo em Redes Sem Fio

Gabriel F. C. Novy<sup>1</sup>, Carlos M. Alves<sup>1</sup>, Fátima L. P. Duarte Figueiredo<sup>1</sup>

<sup>1</sup>Instituto de Informática – Pontifícia Universidade Católica de Minas Gerais (PUC Minas)

Belo Horizonte – MG – Brasil

gabriel.novy@sga.pucminas.br, carlosmbh@gmail.com,  
fatimafig@pucminas.br

**Abstract.** Besides the network games industrial growth, it is necessary a traffic analysis. This work presents an OpenArena multiplayer game simulation model. It was composed by real data collected over a Wi-Fi network and by the simulation modeling. The simulations reflected the data collected and results analyzed to different scenarios. Applications like FTP are more sensible to the network game concurrence. HTTP and others are not so sensible like FTP, as simulation results show. From the simulation model it is possible to predict the game impact over the network. The model can be configured to other applications, games and networks.

**Resumo.** Tendo em mente o espantoso crescimento da indústria de jogos em rede, torna-se necessário um mecanismo de análise do tráfego gerado por esse tipo de aplicação. Este trabalho apresenta um modelo de simulação do jogo multiplayer OpenArena que pode ser utilizado para testes sem a necessidade de um ambiente real. O trabalho envolveu coleta de dados do jogo em um ambiente real Wi-Fi, modelagem e simulação refletindo os dados coletados e análise de resultados frente a diferentes cenários. Os resultados indicam que um número reduzido de aplicações de jogo FPS não degrada o desempenho da rede. Outras aplicações, tais como FTP, são mais sensíveis à concorrência do jogo que outras, como, por exemplo, HTTP. A partir do modelo de simulação, pode-se prever o impacto que jogos eletrônicos podem causar em redes sem fio. O modelo pode ser configurado para outros tipos de jogos, aplicações e redes.

## 1. Introdução

A indústria brasileira de jogos eletrônicos vem crescendo, com profissionais capacitados em empresas que produzem *softwares* para jogos eletrônicos [AbraGames 2009]. As interações entre jogos que utilizam dispositivos móveis tais como celulares, *hand-helds*, *palm tops* e *notebooks* vêm se tornando cada vez mais comuns. As atuais tecnologias de rede para a comunicação desses dispositivos são Bluetooth, Wi-Fi, WIMAX e redes de celulares. O tráfego gerado por jogos pode ter um elevado impacto nessas redes. Para impedir a degradação da rede, é necessário conhecer as demandas dos jogos eletrônicos no que se refere à largura de banda, tempo de resposta, utilização de recursos, dentre outros parâmetros.

As gerações atuais de consoles e dispositivos móveis suportam jogos *multiplayer* em redes de padrão IEEE 802.11. Com isso deve-se estudar o

comportamento desses aplicativos e a interferência deles em outras aplicações concorrentes na rede [Claypool 2005]. Modelos de tráfego na *internet* provenientes de aplicações de jogos *multiplayer* vem sendo estudados há alguns anos. A alta troca de informações, característica dos jogos de tiro em primeira pessoa, os famosos FPS (*First Person Shooter*), são particularmente interessantes no estudo referente ao tráfego gerado e na qualidade de serviço esperada [Branch and Armitage 2006]. Uma parte significativa do tráfego da internet é gerada por jogos eletrônicos. O tráfego gerado pelos jogos é de 3 a 4% do tráfego total de um *backbone* [Färber 2002]. Como essas aplicações possuem alta interação e exigem da rede respostas rápidas, é importante estudar o comportamento desse tráfego e propor alternativas e soluções para eventuais problemas relacionados à degradação da rede. O conhecimento prévio do comportamento dos jogos eletrônicos possibilita a criação de uma estrutura de rede que poderá atender, de maneira satisfatória, as necessidades de comunicação dos usuários dos jogos sem afetar o tráfego de outras aplicações.

Atualmente, as redes locais sem fio são bastante populares. Padronizadas pelo IEEE 802.11 [802.11 2007], estão em ambientes domésticos, corporativos e acadêmicos. De acordo com [Carrig, Defieffe and Murphy 2007], essas redes sem fio são adequadas para aplicações que exigem reduzido tempo de resposta e que possuem rigorosos requisitos em relação ao atraso.

Este trabalho propõe a criação de um modelo simulação como sugerido por [Branch and Armitage 2006]. Esse modelo foi criado a partir de coletas de dados reais em uma rede sem fio. Em seguida foram realizadas simulações de sobrecarga na rede para se avaliar o desempenho de aplicações do jogo, HTTP, FTP e *streaming*. Com base nos resultados foi analisado o impacto que um jogo do gênero FPS, o jogo OpenArena, pode causar em uma rede 802.11g e em outras aplicações. O inverso também foi avaliado: qualidade do jogo com aumento de carga na rede. A motivação do trabalho foi tentar responder a duas questões: qual é o impacto que os jogos podem causar em outras aplicações em execução nas redes sem fio e qual o impacto que outras aplicações na rede podem causar em um jogo. Para que se pudesse responder a essas questões, testes reais e simulações foram conduzidos para se modelar e analisar o comportamento do OpenArena em uma rede Wi-Fi.

O conteúdo deste texto está organizado da seguinte forma: a seção 2 apresenta os conceitos necessários para o entendimento do trabalho. Na seção 3 é apresentada uma descrição da metodologia adotada. A seção 4 apresenta as simulações e os resultados das simulações. A seção 5 apresenta as conclusões e os trabalhos futuros.

## 2. Fundamentação Teórica e Trabalhos Relacionados

Diversos trabalhos analisam e propõem modelos para o comportamento de partidas em rede de jogos de tiro em primeira pessoa [Färber 2002], [Lang, Branch and Branch 2004], [Claypool 2005], [Branch and Armitage 2006], [Oliver, Miller and Allison 2010]. Diversos aspectos tais como latência, vazão da rede e perda de pacotes devem ser considerados na avaliação do comportamento de um jogo em partidas em rede.

Identificar o comportamento de geração de tráfego por um jogo em uma rede pode ajudar a identificar os requisitos necessários para a garantia de qualidade do jogo e o possível impacto que tal jogo venha a causar em outras aplicações na rede. Em [Färber 2002] é descrito o comportamento do tráfego de um jogo do gênero FPS. Os clientes do

jogo enviam, com intervalos regulares, dados para o servidor, informando suas posições e *status* do personagem. O servidor, por sua vez, analisa os dados recebidos, agrupa esses dados e envia uma rajada de mensagens para todos os clientes. Assim, os clientes podem atualizar seu cenário com as informações dos seus adversários. Os intervalos de envio de dados do servidor e do jogador foram analisados por [Lang, Branch and Branch 2004]. Nestas análises verificou-se que, independente do número de jogadores conectados, o servidor envia rajadas de informações a cada 50 ms e que o tamanho do pacote enviado pelo servidor depende do número de jogadores conectados. Já os clientes, enviam pacotes em tempos regulares de aproximadamente 10.75 ms e o tamanho do pacote segue uma distribuição normal, que não varia, independentemente do número de jogadores conectados ao jogo.

Outro aspecto a ser avaliado em uma partida de jogo em rede é a largura de banda consumida pelo jogo. Segundo [Oliver, Miller and Allison 2010] jogos FPS necessitam de, aproximadamente, 67 Kbps de largura de banda por cada aplicação de jogo.

O comportamento de um jogo pode ser constante, durante toda a partida em rede, ou variar quando jogado em plataformas diferentes como, por exemplo, consoles de fabricantes diferentes. Em [Claypool 2005] é realizado um estudo comparando o comportamento dos consoles Nintendo DS e o Sony PSP em uma rede sem fio IEEE 802.11. Avaliando o comportamento do tráfego de vários gêneros de jogos, o autor identificou várias etapas envolvidas no processo execução de uma partida *multiplayer* em rede. Tais etapas são: a etapa sincronização, a etapa da partida propriamente dita e por último a etapa de mudança de cenários. Identificou-se que etapas diferentes possuem consumo de banda diferente. No entanto, é importante ressaltar que no trabalho realizado em [Claypool 2005] é feita uma análise de jogos diferentes e de gêneros diferentes, em cada console, porém não avalia um mesmo jogo nos dois consoles. No entanto, identificar se jogos de um mesmo gênero possuem comportamento semelhante foi um trabalho apresentado por [Ratti, Hariri, and Shirmohammadi 2010], em um *survey* que aborda uma variedade de jogos do tipo FPS. São apresentadas as características de cada jogo, sua versão, tamanho médio de pacote e outras informações que são de suma importância para se entender o comportamento deste gênero de jogo em redes de computadores. Por fim, [Ratti, Hariri, and Shirmohammadi 2010] afirma que diferentes jogos do gênero FPS possuem comportamento semelhante e que os dados analisados se aplicam a uma variedade maior de jogos deste gênero.

O desempenho de um jogo em partidas em rede do ponto de vista do jogador, é apresentado por [Färber 2002], que é a principal referência deste trabalho por apresentar métricas de qualidade para um jogo do gênero FPS. Através de testes em que se aumentou o intervalo de envio de pacotes gradativamente e avaliando-se cada situação do ponto de vista do usuário, [Färber 2002] concluiu que, para que um jogo do gênero FPS possua um desempenho aceitável em partidas *multiplayer*, é necessário que o intervalo de recebimento de pacote de dados seja inferior a 150 ms.

Uma vez avaliado o comportamento de partidas em rede de um jogo é possível criar modelos dos comportamentos de tais jogos, tal como sugerido por [Branch and Armitage 2006]. Com tais modelos é possível estudar o real impacto de um jogo, quando dezenas ou centenas de usuários utilizam uma rede. Este trabalho é

complementar aos da literatura, pois permite a avaliação do impacto de um jogo em uma rede. O modelo desenvolvido neste trabalho foi implementado no Network Simulator 2 que é um simulador de redes de computadores amplamente utilizado no meio acadêmico. Este modelo pode ser facilmente modificado para outros jogos, cenários e redes.

### 3. Descrição da Metodologia

A metodologia do trabalho foi desenvolvida em três etapas: coleta de dados reais, simulação e validação dos experimentos e simulação de novos cenários. A seguir cada etapa será descrita com seus procedimentos detalhados.

#### 3.1. Descrição dos Ambientes de Coleta de Dados Reais

Para a coleta de dados reais foram utilizados ao todo sete *notebooks* conectados a uma rede *wireless* configurada especialmente para esse fim (ambiente controlado). A rede foi composta por um *Access Point (AP)* Linksys WRT54G e pelos *notebooks* conectados a esse *AP*. O *AP* utiliza a tecnologia 802.11 b/g que proporciona desempenho nominal teórico de até 54 Mbps e prático de cerca de 22 Mbps [Wang and Refai 2005], [Wijesinha et al. 2005]. A taxa utilizada foi de 21.7 Mbps [Info 2010]. A conexão de internet utilizada foi de 2 Mbps. Para os experimentos, o *AP* foi configurado com protocolo de criptografia WPA2, protocolo MAC 802.11g, canal de comunicação 6 e servidor DHCP habilitado. O jogo utilizado nas coletas foi o OpenArena. O OpenArena é um jogo do gênero FPS, baseado na *engine* do jogo Quake 3, possui código aberto e é multiplataforma.

Utilizando o ambiente controlado, foi criado um ambiente de testes. Nesse ambiente, os dados gerados pelas aplicações que utilizavam a rede foram capturados através da ferramenta Wireshark v. 1.0.6. A ferramenta foi instalada em todos os *notebooks* que participaram dos experimentos 1 e 2, e todo o tráfego gerado por todos os *notebooks* foram capturados para posterior análise.

O experimento 1 foi composto de 6 cenários em que variou-se o número de jogadores de 1 a 6. Ou seja, no cenário 1 havia o servidor do jogo e um jogador, no cenário 2 havia o servidor e dois jogadores e assim por diante até o cenário seis em que havia o servidor do jogo e seis jogadores.

Os cenários do experimento 2 foram compostos por um servidor, seis jogadores e outros usuários, conectados ao ambiente, executando outras aplicações. Mais especificamente, no cenário 1, havia o servidor do jogo, seis jogadores, uma navegação na WEB e um *streaming* de vídeo. No cenário 2, foi acrescentado mais uma navegação na WEB. Por fim, o cenário 3 foi composto pelo servidor do jogo, 6 jogadores, uma navegação na WEB e uma transferência de arquivo entre dois *notebooks* da rede. Os dados coletados foram utilizados na construção dos cenários de simulação e comparação com resultados de simulações.

Os dois experimentos seguiram a mesma estrutura. O que variou entre as fases dos experimentos foi a diversidade de aplicações utilizando a rede. O tempo de coleta dos dados em cada cenário foi de 20 minutos.

### 3.2. Simulação e Validação dos Experimentos

Antes de se criar os códigos de simulação, que foram executados no Network Simulator 2 (NS2) v. 2.34, foi necessário modelar os dados observados nos experimentos reais, a fim de se definir os elementos que compõem um determinado cenário. Nessa subseção são apresentados os diagramas gráficos que representam as configurações das aplicações criadas no NS2. Esses diagramas possibilitaram a criação de uma ferramenta computacional capaz de gerar os *scripts* de simulação que foram utilizados neste trabalho. As aplicações HTTP, FTP e *streaming* foram utilizadas por serem aplicações típicas de usuários de redes sem fio. Os valores utilizados de parâmetros nas configurações das aplicações e do cenários foram obtidos com base na análise dos resultados reais.

Os cenários do experimento 1 foram compostos por um AP, um servidor do jogo e um número de jogadores que varia para cada fase. No cenário de simulação cada elemento da rede foi representado por um nó. O nó que representava o AP foi configurado como AP. O servidor de jogo foi configurado com múltiplos agentes de transporte para envio de dados - Agent/UDP - e múltiplos agentes para recebimento de dados - Agent/Null -, pois, como observado nas coletas, o servidor possui uma conexão com cada um dos jogadores e todos os jogadores se conectam ao servidor. Cada agente de envio do servidor foi conectado a uma aplicação CBR(*Constant Bit Rate*). Os jogadores foram configurados, cada um, com um agente de envio e um agente de transporte. Os agentes de envio dos jogadores foram conectados, cada um, a uma aplicação CBR. Cada agente de envio do servidor foi conectado a um agente de recebimento de um cliente e os agentes de envio dos clientes foram conectados, cada um, a um diferente agente de recebimento do servidor, conforme ilustra a figura 1-a.

Tanto para o agente de transporte quanto para a aplicação do servidor e do jogador, foram necessários configurar dois parâmetros. O primeiro se refere ao intervalo de envio de pacotes por cada nó. As capturas resultaram em vários valores. Um para cada cenário. Por isso, em cada simulação os valores eram sorteados dentre os obtidos nas capturas. O segundo parâmetro diz respeito ao tamanho dos pacotes enviados por cada nó. Foi observado que, nas capturas, a média do tamanho dos pacotes enviados pelos jogadores se mantinha constante, independentemente da quantidade de jogadores nas partidas. No entanto, a média do tamanho dos pacotes enviados pelo servidor aumentava em 12 bytes para cada novo jogador na partida. Assim, os nós jogadores foram configurados com tamanhos de pacotes iguais, que foi a média obtida nas capturas, enquanto o tamanho dos pacotes enviados pelo servidor aumentava em 12 bytes, para cada novo nó jogador na rede.

Como dito anteriormente, as capturas do experimento 2 foram compostas de usuários executando aplicações diversas, em cenários diferentes. Essas aplicações eram navegação na WEB, *streaming* de vídeo e transferência de arquivo, além do jogo do experimento 1.

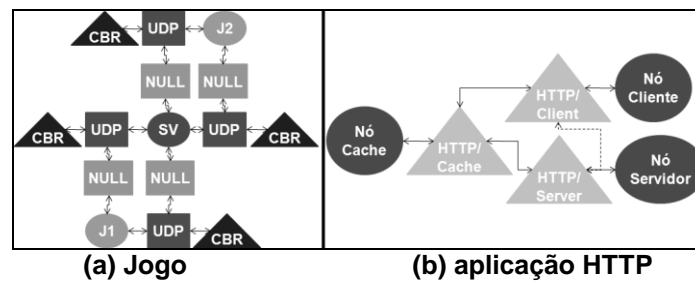


Figura 1. Construção das aplicações do Jogo e HTTP.

A configuração da navegação na WEB foi simulada com a utilização das aplicações Http/Server, Http/Client e Http/Cache. Cada aplicação necessita de um nó que a represente. Assim, para cada aplicação de navegação na WEB acrescentada à simulação eram acrescentados 3 nós. Por serem orientadas a conexão, essas aplicações utilizam o protocolo TCP. A aplicação do servidor (Http/Server) foi conectada à aplicação da cache (Http/Cache), que por sua vez foi conectada à aplicação do cliente (Http/Client). Em seguida, foi aberta uma sessão entre a aplicação do cliente e do servidor. O parâmetro de configuração da aplicação do servidor é o tamanho da página geradora de tráfego, que foi configurado em 135775 bytes. Para a aplicação do cliente, foi necessário informar o intervalo médio entre requisições feitas pelo cliente (11.0 segundos) e o modelo de distribuição dos intervalos entre requisições realizadas pelo cliente (RandomVariable/Exponential). Por último, o tempo médio de vida das páginas (89.99 segundos) e o modelo de distribuição do tempo de vida das páginas (RandomVariable/Exponential) foram utilizados para configurar a aplicação da *cache*. A figura 1-b apresenta como foi construída cada aplicação HTTP nos cenários de simulação.

Cada aplicação *streaming* de vídeo na rede utilizava dois nós. Um nó representava o servidor e o outro o cliente. Cada servidor foi configurado com um agente de envio (Agent/TCP) que era conectado a uma aplicação de *streaming* (Application/SctpApp1). Cada cliente foi configurado com um agente de recebimento (Agent/TCPSink). Por último, o agente de envio de cada servidor foi conectado ao agente recebimento de cada cliente correspondente. No servidor, foram necessários configurar o intervalo médio de envio da aplicação (33.3 ms) e o tamanho dos pacotes do agente de envio (940 bytes). No agente do cliente, foi configurado o tamanho do pacote *ack* (54 bytes). Os valores configurados nos servidores e nos cliente foram extraídos das capturas. A figura 2-a apresenta como foi construída cada aplicação *streaming* nos cenários de simulação.

Por último, cada transferência de arquivo também foi representada utilizando-se dois nós. Um nó representava o servidor de FTP e o outro o cliente. Cada servidor foi configurado com um agente de envio (Agent/TCP) que era conectado a uma aplicação FTP (Application/FTP). Cada cliente foi configurado com um agente de recebimento (Agent/TCPSink). Nesse caso, foi necessário fornecer para a aplicação do servidor a quantidade de pacotes enviados (182400 pacotes) e para o agente de envio o tamanho dos pacotes (1465 bytes). No agente do cliente, também foi configurado o tamanho do pacote *ack* (54 bytes). Os dados utilizados na configuração da simulação também foram extraídos das capturas. A figura 2-b apresenta como foi construída cada aplicação FTP nos cenários de simulação.

Os servidores de navegação na WEB, *streaming* de vídeo e FTP foram posicionados na rede, pois, como não é objetivo do trabalho analisar o tráfego na internet, não há a necessidade de se construir um cenário que simule a integração de uma rede Wi-Fi com a rede de internet.

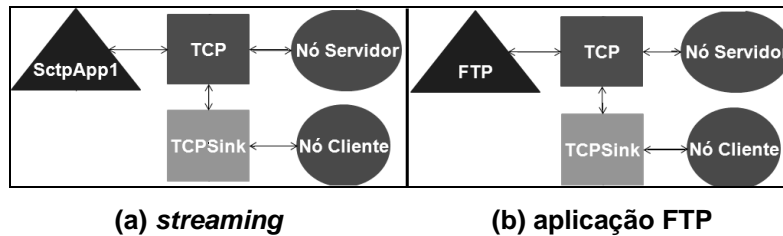


Figura 2. Construção das aplicações *streaming* e FTP.

Para todos os cenários de simulação, a largura de banda efetiva da rede foi ajustada em 21.7 Mbps [Info 2010] e o tempo de simulação foi igual aos das capturas, de 1200 segundos (20 minutos), que foi definido empiricamente, uma vez que é um tempo considerado razoável para partidas de jogo *multiplayer*.

### 3.3. Simulação de Novos Cenários

As simulações de novos cenários foram realizadas em 2 etapas. Na situação 1 foram realizadas simulações individuais para cada aplicação (jogo, HTTP, *streaming* e FTP). Cada cenário era composto de múltiplas aplicações de um mesmo tipo. Ao todo foram 20 cenários, que variavam de 1 a 20 aplicações, para cada tipo de aplicação. A construção de novos cenários se deu da mesma forma que a construção de cenários para validação. A figura 3 apresenta a configuração dos novos cenários de simulação da etapa 1.



Figura 3. Construção de novos cenários de simulação para a etapa 1.

A etapa 2 foi composta de simulações de cenários compostos de várias aplicações. Visando avaliar o impacto do jogo em outras aplicações e vice-versa, a etapa 2 foi realizada seguindo 3 abordagens. Na primeira abordagem o número de jogadores era fixo e variava-se o número de cada aplicação entre 1 e 20. Já na segunda abordagem a quantidade de cada aplicação na rede era fixo (6 para HTTP, 6 para *streaming* e 4 para FTP) e variava-se a quantidade de jogadores entre 1 e 15, em simulações individuais para cada aplicação. A terceira abordagem continha as 4 aplicações em um mesmo cenário. As figuras 4 e 5 apresentam as três abordagens de configuração dos novos cenários de simulação da etapa 2.

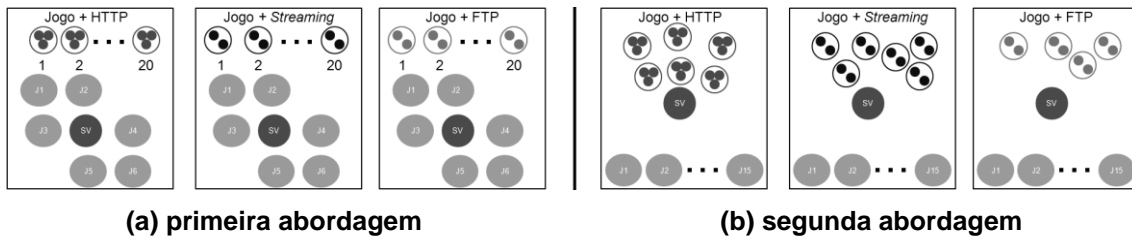


Figura 4. Construção de novos cenários de simulação para a etapa 2.

Para que fosse acrescentado nos cenários de simulação uma variação no comportamento das aplicações que melhor representasse situações reais, os parâmetros utilizados nas configurações eram sorteados entre os valores encontrados nas capturas. Também, para melhor representar um comportamento real, e como observado nas capturas, o tamanho dos pacotes enviados pelo servidor do jogo aumentava em 12 bytes para cada novo jogador na rede.

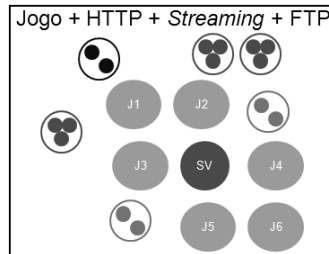


Figura 5. Construção de novos cenários de simulação para a etapa 2 – terceira abordagem.

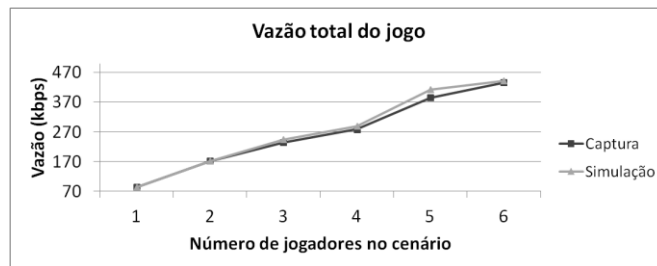
## 4. Resultados

Esta seção apresenta os resultados da validação dos cenários construídos e os resultados de simulações de novos cenários para os experimentos 1 e 2. Ela apresenta os resultados das validações e das simulações de novos cenários dos experimentos 1 e 2. Os parâmetros utilizados para avaliar os resultados foram: vazão por tipo de aplicação, intervalo médio de envio e de recebimento de pacotes, percentual de pacotes perdidos e variação do intervalo de recebimento de pacotes. Com base nesses parâmetros, foi avaliado o desempenho de cada aplicação na rede, separadamente, foram avaliados: (1) jogo, individualmente, e cada aplicação, individualmente, (2) jogo com cada aplicação escolhida, individualmente e (3) separadas e jogos com múltiplas aplicações simultâneas.

### 4.1. Validação dos Experimentos 1 e 2

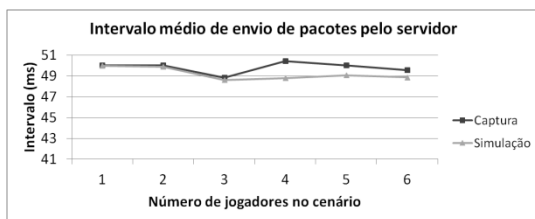
Esta subseção apresenta a comparação dos dados obtidos nas capturas do experimento 1, com os dados obtidos nas simulações dos cenários de validação do experimento 1 e os resultados de simulações de novos cenários para o experimento 1. Como já foi dito, os cenários são compostos somente pela aplicação do jogo e contêm até 20 jogadores. A seguir são apresentados os resultados.



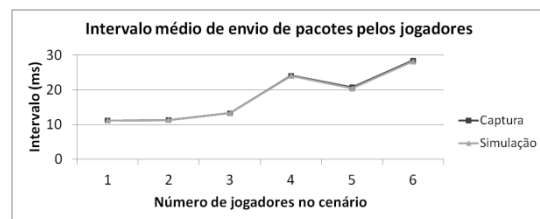


**Figura 6. Comparação entre a vazão total do jogo obtida nas capturas e a vazão total do jogo obtida nas simulações em cenários com até 6 jogadores.**

O gráfico da figura 6 mostra comparação entre a vazão total do jogo obtida nas capturas e a vazão total do jogo obtida nas simulações em cenários com até 6 jogadores. Como mostram os resultados, o comportamento da aplicação do jogo nas simulações foi bastante semelhante ao comportamento do jogo real. Isso mostra que, neste aspecto, o cenário de simulação foi validado.



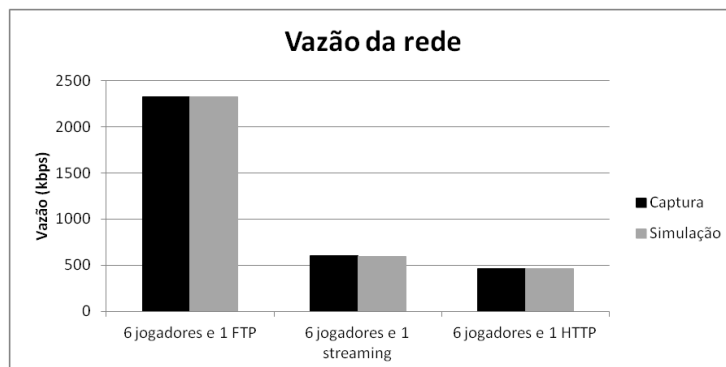
**(a) Servidor para jogadores**



**(b) Jogadores para servidor**

**Figura 7. Comparação do intervalo médio de envio e recebimento de pacotes pelo servidor e pelos jogadores para cenários com até 6 jogadores.**

Nos gráficos da figura 7 são apresentados os intervalos de envio de pacotes para as capturas reais e para a simulação dos cenários de validação com até 6 jogadores. Pode-se perceber que a aplicação do jogo se comportou, na simulação, de forma bastante semelhante ao jogo real. Nota-se que a média dos intervalos de envio de pacotes pelos jogadores varia bastante. Isso ocorre porque, nas capturas reais, havia máquinas com capacidade de processamento de melhor e de pior qualidade. Isso fez com que ocorresse tal variação da média.



**Figura 8. Comparação, por aplicação, entre a vazão total obtida nas capturas e a vazão total obtida nas simulações.**

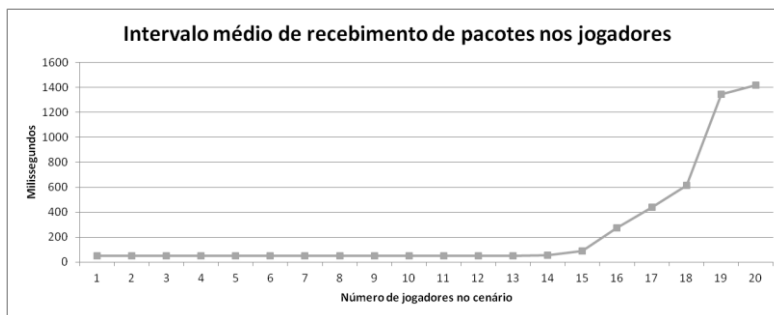
O gráfico da figura 8 compara os dados de vazão total por aplicação, obtido nas capturas, com os dados de vazão total por aplicação encontrada nas simulações. É

possível perceber que os dados estão bastante semelhantes, validando, assim, os cenários criados e as aplicações.

#### 4.2. Simulação de Novos Cenários

Nesta subseção são apresentados os resultados das simulações de novos cenários para as etapas 1 e 2. A seguir são apresentados e analisados os dados.

O gráfico da figura 9 apresenta o intervalo de recebimento de pacotes nos jogadores nos cenários da etapa 1. Nos cenários da etapa 1, o jogo utiliza a rede sem concorrência e apresenta perda de desempenho a partir do cenário composto por 16 jogadores, em que o intervalo de recebimento de pacotes ultrapassa a marca de 150 ms, conforme recomendado por [Färber 2002].



**Figura 9. Intervalo médio de recebimento de pacotes nos jogadores para os cenários da etapa 1.**

A tabela 1 apresenta o intervalo médio de recebimento de pacotes nos jogadores em diferentes cenários da etapa 2. A coluna “Jogadores” indica quantos jogadores estavam presentes no cenário. A coluna “Aplicações” apresenta quantas aplicações de cada tipo estavam presentes no cenário. Como indica os valores das colunas “Jogadores” e “Aplicações”, a quantidade de cada tipo de aplicação no cenário podia ser fixa ou variável. Para as simulações definiu-se 6 o número inicial de jogadores, por ter sido mostrado, em experimentos reais, que esse número não compromete o desempenho da rede. Em cenários em que a quantidade era variável, foram realizadas simulações para cada valor do intervalo. Por último, a coluna “Cenário com intervalo maior que 150 ms” apresenta os cenários a partir de onde o desempenho do jogo se tornou insatisfatório, ultrapassando a marca de 150 ms, conforme recomendado por [Färber 2002]. A aplicação HTTP não interfere no jogo. Em um cenário com 6 jogadores, são necessárias 15 aplicações *streaming* para interferir na qualidade do jogo. A perda de qualidade é percebida em um cenário de 6 jogadores com apenas 8 aplicações FTP. Os dados da tabela mostram, ainda, o impacto específico de 6 aplicações *streaming* e 4 aplicações FTP. Seis aplicações *streaming* começam a interferir no desempenho do jogo quando o número de jogadores chega a 13. Em um cenário composto de até 8 jogadores, a rede só suporta 4 FTPs, sem que o desempenho seja prejudicado. A partir de 4 FTPs, o intervalo de recebimento de pacotes ultrapassa 150 ms.

Tabela 1. Desempenho do jogo nos cenários da etapa 2

Etapa 2		
Jogadores	Aplicações	Cenário com intervalo maior que 150 ms nos jogadores
6	HTTP (1-20)	-
	Streaming (1-20)	15 streamings
	FTP (1-20)	8 FTPs
1-15	HTTP (6)	-
	Streaming (6)	13 jogadores
	FTP (4)	9 jogadores

Com base na análise da tabela 1, é possível notar que a inserção de aplicações *streaming* na rede é impactante na qualidade do jogo, quase que na mesma medida que a inserção de novos jogadores. É possível perceber, também, que a aplicação FTP possui grande impacto na qualidade do jogo, visto que ela demanda altos recursos da rede.

A análise sobre o impacto do jogo em aplicações HTTP foi feita analisando-se a vazão das aplicações HTTP. Em todos os cenários em que havia aplicações HTTP, a vazão média de HTTP ficou em cerca de 110 Kbps. Notou-se, então, que a inserção de jogadores na rede não altera o desempenho de uma aplicação de HTTP, visto que na simulação de um cenário composto de apenas um HTTP, a vazão média também foi próxima de 110 Kbps.

A avaliação do impacto do jogo sobre aplicações *streaming* foi através da análise da vazão média e do *jitter*. A vazão em cenários compostos somente por aplicações *streaming* se manteve em uma média de cerca de 240 Kbps. Em cenários em que a aplicação *streaming* teve a concorrência do jogo, a vazão média também foi próxima 240 Kbps. Por isso, é possível perceber que o jogo, nos cenários simulados, não interfere na vazão de aplicações *streaming*. O *jitter*, em todos os cenários, se manteve abaixo de 1ms, considerado satisfatório.

Por último, houve uma avaliação do desempenho de aplicação FTP. Como mostra o gráfico da figura 10, o incremento de aplicações FTP na rede interfere em aplicações já presentes, fazendo com que o percentual de perda de pacotes aumente.

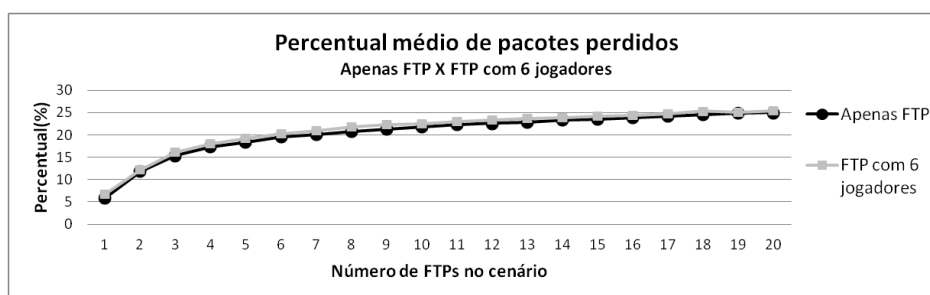


Figura 10. Porcentagem média de pacotes perdidos da aplicação FTP.

O gráfico da figura 10 também mostra o impacto que 6 jogadores tem em cenários com várias aplicações FTP. Os resultados foram semelhantes aos cenários onde as aplicações FTP não concorriam com o jogo. O percentual médio de pacotes perdidos das aplicações FTP com concorrência de 6 jogadores aumentou em menos de 1%. Isso mostra que o fator impactante no desempenho de aplicações FTP é a concorrência com as próprias aplicações FTP. Tal comportamento também foi observado nos cenários com múltiplas aplicações visto que um maior impacto na perda de pacotes das aplicações FTP só ocorria com o acréscimo de novas aplicações FTP.

## 5. Conclusões e Trabalhos Futuros

Este trabalho propôs a criação de um modelo de simulação para avaliação do impacto que um jogo causa em redes sem fio. A construção do modelo foi realizada com base em dados reais obtidos em capturas de tráfego. Inicialmente, foram simulados os mesmos cenários dos testes reais. O modelo foi validado através da comparação de resultados de simulação com resultados reais. Após a validação, novos cenários foram construídos, possibilitando, assim, a realização de simulações de ambientes com maiores quantidades de jogadores e outras aplicações.

Conforme mostraram resultados de simulação, a partir de 15 jogadores de jogos de tiro em primeira pessoa, o atraso médio por pacote do jogo fica superior aos 150 ms recomendados por [Färber 2002], comprometendo a qualidade do jogo. Este resultado aponta para uma capacidade máxima de 15 jogadores FPS com qualidade, uma vez que, como apresentado por [Ratti 2010], diversos jogos do gênero FPS possuem comportamento semelhante. A partir de 16 jogadores, conforme mostrado pelos resultados, o desempenho do jogo fica insatisfatório.

Com base nos resultados obtidos, pôde-se analisar o comportamento de uma rede sem fio frente à quantidade de jogadores e de outras aplicações. Foi avaliado, também, o impacto que aplicações do tipo HTTP, *streaming* e FTP têm em um jogo FPS na rede sem fio. Os resultados obtidos mostraram que a uma quantidade elevada de aplicações HTTP pouco interfere na qualidade do jogo. Aplicações *streaming* causam impacto semelhante ao gerado pelos próprios jogadores na rede. Percebeu-se que, quando há mais jogadores (acima de 13), apenas 6 aplicações *streaming* já degradam o desempenho do jogo. Com poucos jogadores (até 6), 15 aplicações *streaming* degradam a rede. Aplicações FTP se mostraram as que mais degradam a qualidade da rede. Percebeu-se que, quando há mais jogadores (acima de 9), apenas 4 aplicações FTP já degradam o desempenho do jogo. Com poucos jogadores (até 6), 8 aplicações FTP degradam a rede.

Mostrou-se, também, impacto do jogo em aplicações do tipo HTTP, *streaming* e FTP. Percebeu-se que uma quantidade pequena de jogadores não causa grande impacto em outras aplicações. Com até 6 jogadores na rede, não houve degradação de qualidade nas outras aplicações. Quando o número de jogadores aumenta, há a possibilidade de degradação no desempenho da rede, dependendo do número e do tipo das aplicações concorrentes. A partir de 15 jogadores, a degradação é concreta e pode ser percebida no aumento do atraso médio dos pacotes do próprio jogo. No entanto, este trabalho não abordou todos os cenários possíveis e recomenda-se que novos cenários sejam avaliados. Também é importante ressaltar que este trabalho abordou somente um jogo do gênero FPS. Avaliar jogos de outros gêneros também é importante, visto que jogos de gêneros diferentes podem possuir comportamentos diferentes. Outro aspecto importante a ser pesquisado é o desempenho do jogo e de outras aplicações em outros tipos de rede sem fio como, por exemplo, redes 3G e Wi-MAX, pois redes diferentes possuem recursos diferentes. Por último, pode ser analisado, também, o comportamento de jogos de consoles como Nintendo DS e Play Station 3.

## Referências

- 802.11 (2007). “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”. Disponível em: <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>.
- AbraGames (2009). Novembro. Acessado em 20/11/2009. Disponível em: [www.abragames.org/docs/Abragames-Pesquisa2008.pdf](http://www.abragames.org/docs/Abragames-Pesquisa2008.pdf).
- Branch, P., Armitage, G. (2006). “Extrapolating server to client ip traffic from empirical measurements of first person shooter games”. ACM Press. Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games , ISBN:1-58113-493-2. Braunschweig, Germany.
- Carrig, B., Defieffe, D. and Murphy, J. (2007). “A non-elevated scheduling solution for first person shooter games in ieee 802.11 network”. Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE, n. 10.1109/GLOCOM.2007.1000, p. Page(s):5277 - 5282, Novembro. Disponível em: <http://ieeexplore.ieee.org/>.
- Claypool, M. (2005). “On the 802.11 Turbulence of Nintendo DS and Sony PSP Hand-held Network Games” In: Netgames'05 10-11 October.
- Färber, J. (2002). “Network Game Traffic Modelling” In: Proceedings of the 1st workshop on Network and system support for games. p.53-57, 16 Abril, Braunschweig, Germany.
- Info abril (2010). Info Lab Web Site. 2010. Acessado em 12.04.2010. Disponível em: <http://info.abril.com.br/reviews/hardware/redes/seguranca-e-o-forte-dowrt54g.shtml>.
- Lang, T., Branch, P. and Armitage, G. (2004). “A Synthetic Traffic Model for Quake3” In: ACE'04 3-5 June.
- Oliver, I., Miller, A. and Allison, C. (2010) “Virtual worlds, real traffic: Interaction and adaptation”. ACM Proceedings of the first annual ACM SIGMM conference on Multimedia systems. ISBN: 978-1-60558-914-5.
- Ratti, S., Hariri, B. and Shirmohammadi, S. (2010). “A survey and analysis of first person shooter gaming traffic on the internet”. Internet Computing, IEEE. ISBN: 978-963-9799-36-3.
- Wang, T., and Refai, H. (2005). “Network performance analysis on ieee 802.11g with different protocols and signal to noise ratio values.” In Wireless and Optical Communications Networks, 2005. WOCN 2005. Second IFIP International Conference on, 29 – 33.
- Wijesinha, A. L., Song, Y.-T., Krishnan, M., Mathur, V., Ahn, J., and Shyamasundar, V. (2005). “Throughput measurement for udp traffic in an ieee 802.11g wlan.” In Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks, IEEE Computer Society, Washington, DC, USA, 220–225.