

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Programa de Pós-Graduação em Informática

ANÁLISE E MODELAGEM DE TRÁFEGO DE JOGOS *FPS*
EM REDES 802.11 EM AMBIENTES *INDOOR*

CARLOS MARCOS ALVES

Belo Horizonte

2010

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

CARLOS MARCOS ALVES

**ANÁLISE E MODELAGEM DE TRÁFEGO DE JOGOS *FPS*
EM REDES 802.11 EM AMBIENTES *INDOOR***

Dissertação apresentado ao Programa de Pós-Graduação em Informática como requisito parcial para qualificação ao Grau de Mestre em Informática pela Pontifícia Universidade Católica de Minas Gerais.

Orientadora: Fátima de Lima P. D. Figueiredo

Belo Horizonte

2010

FICHA CATALOGRÁFICA

Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

A474a	<p>Alves, Carlos Marcos Análise e modelagem de tráfego de jogos FPS em redes 802.11 em ambientes indoor / Carlos Marcos Alves. – Belo Horizonte, 2010. 71f. : il.</p> <p>Orientador: Fátima de Lima Procópio Duarte Figueiredo Dissertação (Mestrado) – Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-graduação em Informática. Bibliografia.</p> <p>1. Jogos eletrônicos – Teses. 2. Simulação (Computadores digitais). 3. Wireless LANs. I. Figueiredo, Fátima de Lima Procópio Duarte Figueiredo. II. Pontifícia Universidade Católica de Minas Gerais. III. Título.</p>
CDU: 681.3.092	

Bibliotecário: Fernando A. Dias – CRB6/1084



PUC Minas
Programa de Pós-graduação em Informática

FOLHA DE APROVAÇÃO

Análise e Modelagem de Tráfego de Jogos FPS em Redes 802.11 em Ambientes Indoor

CARLOS MARCOS ALVES

Dissertação defendida e aprovada pela seguinte banca examinadora:

Fátima de Lima Procópio Duarte Figueiredo

Prof^ª. Fátima de Lima Procópio Duarte Figueiredo - Orientadora (PUC Minas)
Doutora em Ciências da Computação - UFMG

Keiko V. C. Fonseca

Prof^ª. Keiko Verônica Ono Fonseca (UTFPR)
Doutora em Engenharia Elétrica - UFSC

Ricardo Poley Martins Ferreira

Prof. Ricardo Poley Martins Ferreira - (UFMG)
Doutor em Ciência da Computação - UFMG

Luis Enrique Zárate Galvez

Prof. Luis Enrique Zárate Galvez (PUC Minas)
Doutor em Engenharia Metalúrgica e de Minas, UFMG

Belo Horizonte, 06 de agosto de 2010.

*A minha adorável esposa.
Pelo amor, carinho e paciência.*

AGRADECIMENTOS

Pais

Aos meus pais, que tão grandiosamente me guiaram e transmitiram seus valores e idéias.

Esposa

À minha adorável esposa. Dôra, pelo amor, carinho e paciência.

Filho

Ao meu filho. Vítor, por entender os momentos de ausência, pelos sorrisos e carinhos.

Orientadora

À minha orientadora Fátima de Lima Procópio, pela direção, esforço, esclarecimento e boas idéias.

Família

Aos meus familiares, pela torcida, força e esperança.

Datapuc

Aos colegas de trabalho.
Minha segunda família.

Amigos

Aos amigos do mestrado e do *Grupo de Pesquisa de Computação Móvel da PUC-MINAS*, especialmente ao Gabriel Novy, pelo apoio e dedicação.

Professores

Aos professores do mestrado, pelos sábios ensinamentos transmitidos.

Colaboradores

Aos amigos que gentilmente cederam seu precioso tempo para participar dos experimentos.

RESUMO

Os jogos sempre fascinaram o homem. Com o avanço da tecnologia, o ato de jogar tomou proporções inimagináveis. O crescimento da indústria de jogos eletrônicos, envolvendo interatividade entre jogadores propiciada pelas redes de computadores, implica em ambientes de jogo com dezenas, centenas e até milhares de pessoas conectadas. Os jogos eletrônicos são classificados em tipos e gêneros. Cada gênero tem suas próprias características e requisitos no que se refere à conectividade, recursos de *hardware* e número de usuários conectados.

Tendo em mente o espantoso crescimento do universo dos jogos eletrônicos, torna-se necessário um mecanismo de análise do tráfego gerado por esse tipo de aplicação. Criar um modelo que reflita o comportamento dos jogos em redes é um desafio na busca de garantias de qualidade de serviço (*QoS*) para essas aplicações, pois é importante saber qual será o impacto que os jogos eletrônicos irão causar nas redes. Por isso, este trabalho propõe a modelagem, através de simulações, de um tipo de jogo em uma rede sem fio. O tráfego gerado pelo gênero *First Person Shooter (FPS)* foi analisado e modelado. O gênero *FPS* foi escolhido porque, nesse, há uma alta taxa de transferências de informações. O jogo selecionado foi o *OpenArena*, que possui todas as características dos famosos jogos *multiplayers*. A escolha se deve ao fato do *OpenArena* pertencer ao gênero *FPS*, ser um jogo de distribuição gratuita, possuir código fonte aberto e possuir compatibilidade com diversas plataformas, como por exemplo, *Microsoft Windows*, *Linux*, *Mac* e *OpenBSD*.

O trabalho foi dividido em três etapas: coleta de dados do jogo em um ambiente real, modelagem e simulação refletindo os dados coletados, análise de resultados frente a diferentes cenários.

Os resultados indicam que poucos jogadores não degradam o desempenho da rede e aplicações, tais como *FTP*, são mais sensíveis à concorrência do jogo que outras, como *HTTP*.

A partir do modelo de simulação, pode-se prever o impacto que jogos eletrônicos de computador podem causar em redes sem fio. O modelo pode ser configurado para outros tipos de jogos, aplicações e de redes.

Palavras-chave: Jogos eletrônicos, Rede sem fio, Modelo.

ABSTRACT

The games always fascinated human kind. With the advancement of technology, gaming took unimaginable proportions. The growth of the electronic game industry, involving interaction between players possible by computer networks, involves gaming environments with tens, hundreds or even thousands of people connected. The games were classified into types and genres. Each genre has its own characteristics and requirements with regard to connectivity, hardware resources and number of connected users.

Bearing in mind the astonishing growth of the universe of electronic games, it is necessary a mechanism to review the traffic generated by this type of application. Create a model that reflects the behavior of the games in networks is a challenge in seeking guarantees of quality of service (*QoS*) for these applications, it is important to know what impact that electronic games will cause the networks. Therefore, this paper proposes the modeling, through simulations, of a type of game on a wireless network. The traffic generated by the genre First Person Shooter (*FPS*) was analyzed and modeled. The *FPS* genre was chosen because, in that there is a high rate of transfer of information. The game was selected *OpenArena*, which has all the features of the famous multiplayer games. The choice is because of *OpenArena* belong to the *FPS* genre, be a set of free distribution have open source and have compatibility with various platforms, such as Microsoft Windows, Linux, OpenBSD and Mac.

The work was divided into three stages: data collection of the game in a real environment, modeling and simulation reflecting the collected data, analysis of results to different scenarios.

The results indicate that few players will not degrade network performance and applications such as FTP are more sensitive to competition of the game than others, such as HTTP.

From the simulation model can predict the impact that computer games can have on wireless networks. The model can be configured for other types of games, applications and networks.

keywords: Games, Wireless network , Model.

LISTA DE FIGURAS

FIGURA 1	Comparação entre dois clássicos do videogame - PAGMAN	20
FIGURA 2	Comparação entre dois clássicos do videogame - ENDURO	20
FIGURA 3	Experimento 1. Um servidor e seis clientes.	31
FIGURA 4	Esboço do modelo criado no <i>NS2</i> .	34
FIGURA 5	Vazão média do jogo para cada fase do experimento 1.	36
FIGURA 6	Vazão média das fases do experimento 1.	37
FIGURA 7	Jogadores VS Percentagem de utilização da rede.	38
FIGURA 8	Vazão média das fases do experimento 2.	40
FIGURA 9	Percentual de utilização da rede para as fases do experimento 2.	41
FIGURA 10	Esquema gráfico de construção do cenário de simulação do experimento 1.	43
FIGURA 11	Esquema gráfico de construção da aplicação HTTP.	44
FIGURA 12	Esquema gráfico de construção da aplicação <i>streaming</i> .	45

FIGURA 13	Esquema gráfico de construção da aplicação FTP.	46
FIGURA 14	Intervalo de envio de pacotes (SERVIDOR) - Experimentos VS Simulações.	48
FIGURA 15	Intervalo de envio de pacotes (JOGADOR) - Experimentos VS Simulações.	48
FIGURA 16	Quantidade de pacotes enviados (SERVIDOR) - Experimentos VS Simulações.	49
FIGURA 17	Quantidade de pacotes enviados (JOGADORES) - Experimentos VS Simulações.	50
FIGURA 18	Consumo de banda - Experimento 1 VS Simulações.	50
FIGURA 19	Consumo de banda - Experimento 2 VS Simulações.	51
FIGURA 20	Vazão total do jogo de 1 jogador até 20 jogadores.	52
FIGURA 21	Intervalo de envio de pacotes do servidor e recebimento no jogador.	52
FIGURA 22	Intervalo de envio de pacotes do jogador e recebimento no servidor.	53
FIGURA 23	Quantidade de pacotes perdidos.	53
FIGURA 24	Vazão média da aplicação até 20 HTTP.	55
FIGURA 25	Vazão média da aplicação até 20 FTP.	56

FIGURA 26	Percentual de perda de pacotes da aplicação até 20 FTP.	56
FIGURA 27	Vazão média de 6 jogadores e da aplicação até 20 HTTP.	57
FIGURA 28	Vazão média de 6 jogadores e da aplicação até 20 <i>streaming</i>	58
FIGURA 29	Intervalo de envio de pacotes do servidor e recebimento no jogador.	58
FIGURA 30	Intervalo de envio de pacotes do jogador e recebimento no servidor.	59
FIGURA 31	Vazão média de 6 jogadores sozinhos na rede e 6 jogadores e <i>streaming</i> .	59
FIGURA 32	Vazão média de 6 jogadores e da aplicação até 20 FTP.	60
FIGURA 33	Intervalo de envio de pacotes do servidor e recebimento no Jogador.	60
FIGURA 34	Vazão média de 6 jogadores sozinhos na rede e 6 jogadores e FTP. .	61
FIGURA 35	Vazão média de 6 aplicações HTTP e jogadores até 15.	62
FIGURA 36	Intervalo de envio do servidor e recebimento no jogador - 6 HTTP e jogadores até 15.	62
FIGURA 37	Vazão média de 6 aplicações <i>streaming</i> e jogadores até 15.	62
FIGURA 38	Intervalo de envio do servidor e recebimento no jogador - 6 <i>streaming</i> e jogadores até 15.	63
FIGURA 39	Vazão média de 4 aplicações FTP e jogadores até 15.	63

FIGURA 40	Intervalo de envio do servidor e recebimento no jogador - 4 FTP e jogadores até 15.	64
FIGURA 41	Percentual de perda de pacotes para a aplicação FTP.	64
FIGURA 42	Vazão da aplicação HTTP para os diversos cenários.	66
FIGURA 43	Percentual de perda de pacotes para a aplicação FTP para os diversos cenários.	66
FIGURA 44	Intervalo de envio de pacotes do servidor e recebimento no jogador.	67

LISTA DE TABELAS

TABELA 1	Especificações dos jogos segundo seu gêneros.	22
TABELA 2	Cenário para coleta de dados - Experimento 1 (Experimentos reais).	29
TABELA 3	Cenário para coleta de dados - Experimento 2 (Experimentos reais).	30
TABELA 4	Detalhamento das fases dos experimentos 1 (Somente o jogo sendo executado).	30
TABELA 5	Detalhamento das fases dos experimentos com outras aplicações (Experimento 2).	31
TABELA 6	Vazão de todas as fases dos experimentos 1.	37
TABELA 7	Fases do Experimento 2 - Jogadores, servidor do jogo e outras aplicações.	39
TABELA 8	Detalhamento dos dados capturados para cada fase do experimento 2.	41
TABELA 9	Intervalo de envio (SERVIDOR) - Experimento 1.	49
TABELA 10	Intervalo de envio (JOGADOR) - Experimento 1.	49
TABELA 11	Cenários da fase 4, jogo e outras aplicações.	64

LISTA DE SIGLAS

ACK *Acknowledgement*

AMD *Advanced Micro Devices, Inc.*

AP *Access Point*

CBR *Constant Bit Rate*

CTF *Capture the Flag*

CRC *Centro de Recursos Computacional*

DHCP *Dynamic Host Configuration Protocol*

FPS *First Person Shooter*

FTP *File Transfer Protocol*

GHz *Giga Hertz*

GB *Giga Bytes*

HD *Hard Disk*

IEEE *Institute of Electrical and Electronics Engineers*

IA *Inteligência Artificial / Artificial Intelligence*

IP *Internet Protocol / Protocolo de Interconexão*

Kbps *Kilobits por segundos*

LAN *Local Area Network / Rede Local*

MAC *Media Access Control*

MB *Mega Bytes*

Mbps *Mega bits por segundos*

MHZ *Mega Hertz*

MIDI *Musical Instrument Digital Interface*

MIT *Massachusetts Institute of Technology*

MMORPG *Massively Multiplayer Online Role Playing Game*

MS *Milissegundos*

NRT *Non Real Time*

NS2 *Network Simulator 2*

P2P *Peer-to-Peer*

PC *Personal Computer*

PhD *Doctor of Philosophy*

PS2 *Play Station 2*

PS3 *Play Station 3*

PSP *Play Station Portable*

QoE *Quality of Experience*

QoS *Quality of Service*

RAM *Random-Access Memory*

RPG) *Role Playing Game*

RTS *Real Time Strategy*

SSID *Service Set Identification*

TCP *Transmission Control Protocol / Protocolo de Controle de Transmissão*

UDP *User Datagram Protocol*

WIFI *wireless fidelity*

WIMAX *Worldwide Interoperability for Microwave Access / Interoperabilidade Mundial para Acesso de Micro-ondas*

WLAN *Wireless Local Area Network / Rede Local Sem Fio*

WPA2 *Wi-Fi Protected Access 2*

SUMÁRIO

1	INTRODUÇÃO	16
2	FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	19
2.1	Jogos	19
2.2	Rede Wireless	24
3	DESCRIÇÃO DA METODOLOGIA E DO MODELO PROPOSTO ..	27
3.1	Ferramenta de captura de dados	28
3.2	Descrição do ambiente de testes	28
3.3	Modelo Proposto	32
4	EXPERIMENTOS E RESULTADOS	35
4.1	Experimento 1	35
4.2	Experimento 2	39
5	MODELO DE SIMULAÇÃO E RESULTADOS	42
5.1	Modelagem dos experimentos para o <i>NS2</i>	42
5.2	Gerador de <i>Scripts</i> / Analisador	46
5.3	Validação das Simulações - Experimento 1	48
5.4	Validação das Simulações - Experimento 2	50
5.5	Simulações de novos cenários - Experimento 1	51
5.6	Simulações de novos cenários - Experimento 2	53
5.6.1	<i>Fase 1 - Somente Aplicação</i>	55
5.6.2	<i>Fase 2 - Simulações com 6 Jogadores e Aplicações</i>	57

5.6.3	<i>Fase 3 - Simulações Aplicação Fixa e Jogadores até 15</i>	61
5.6.4	<i>Fase 4 - Simulações de Cenários Mistos</i>	64
6	CONCLUSÕES E TRABALHOS FUTUROS	68
6.1	Experimentos Reais	68
6.2	Modelo de Simulação	68
6.3	Trabalhos Futuros	69
	REFERÊNCIAS	70

1 INTRODUÇÃO

A indústria de jogos de computadores faturou US\$ 22 bilhões nos Estados Unidos em 2008, segundo (NPDGROUP, 2009) (instituição criada em 1967 para realizar pesquisa de mercado nos Estados Unidos da América). 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 novas tecnologias de rede que podem ser citadas são: *bluetooth*, *Wi-Fi*, *WIMAX* e redes de celulares, para conectividade desses aparelhos. O tráfego gerado por esses dispositivos podem ter um impacto dramático nas redes. Para impedir a degradação da rede, será necessário conhecer as demandas dos jogos eletrônicos que utilizam esses dispositivos, no que se refere à largura de banda, tempo de resposta, utilização de recursos, dentre outros parâmetros.

Com o crescente uso de jogos nas redes sem fio, as novas gerações de console que suportam jogos *multiplayers*¹ sobre as redes de padrão IEEE 802.11, fica claro que devemos estudar o comportamento desses aplicativos para que possamos prever o comportamento dos mesmos em uma rede de computadores (CLAYPOOL, 2005).

Já na internet, modelos de tráfego 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 PHILIP; ARMITAGE, 2006).

Neste trabalho, para avaliação do impacto que um jogo pode causar nas redes *Wi-Fi*, foi selecionado um dos jogos mais populares na internet. O gênero escolhido foi o de tiro em primeira pessoa (*FPS*). O jogo selecionado foi o *OpenArena*. Este jogo foi escolhido por ser de código aberto, *multiplayer* e possuir todos os requisitos de um jogo

¹jogos com vários jogadores conectados a um servidor.

de arquitetura cliente/servidor. Outra grande vantagem é que sua distribuição é livre, ou seja, o jogo pode ser baixado gratuitamente. Desta forma, pode-se realizar pesquisas relacionadas à área de jogos utilizando um jogo que tem as mesmas características dos famosos jogos *FPS* sem a necessidade de adquirir diversas licenças.

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*² (FARBER, 2002). Como essas aplicações têm muita 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 às necessidades de comunicação dos usuários dos jogos de computadores sem afetar o tráfego de outras aplicações.

Atualmente, as redes sem fio vêm se popularizando. Padronizadas pelo IEEE 802.11 (IEEE-STANDARD, 2010), estão se espalhando rapidamente, seja em ambientes domésticos, corporativos ou acadêmicos. O grande aumento na utilização de redes sem fio (*WLAN*), com o propósito de jogar, traz, intrinsecamente, uma questão. As redes *WLAN* são adequadas para aplicações que exigem baixo tempo de resposta e que possuem rigorosos requisitos em relação ao atraso (CARRIG BRIAN; DEFIEFFE, 2007). Para as redes sem fio (*WLAN*), foco deste trabalho, pode-se aplicar as políticas de qualidade de serviço (*QoS*) que foram implementadas no padrão 802.11e (KINICK JAMES; CLAYPOOL, 2008). A utilização deste padrão nos permite realizar configurações em nosso ambiente que possibilite a priorização de determinado tipo de aplicação.

Neste trabalho, será analisado o impacto que o jogo *OpenArena* pode causar em uma rede 802.11g e em outras aplicações o inverso também será avaliado: qualidade do jogo com aumento de carga à rede.

Este trabalho teve como principal motivação responder a duas perguntas:

1. Qual é o impacto que os jogos de computadores podem causar em outras aplicações em execução nas redes sem fio?
2. Para os jogadores, o desempenho do jogo é satisfatório quando o mesmo concorre com o tráfego de outras aplicações em uma rede sem fio?

Para responder a essas perguntas, testes reais e simulações foram desenvolvidos,

²Traduzido: Espinha dorsal. É a central de alto desempenho que faz a interligação de todas as redes, geralmente o ponto de saída.

para se modelar e analisar o comportamento do *OpenArena* em uma rede *Wi-Fi*.

A simulação tem um importante papel em pesquisas que necessitem verificar o comportamento de determinado elemento ao longo do tempo ou em relação ao número de indivíduos. Neste contexto, a simulação se faz necessária, pois avaliará o impacto que jogos (*FPS*) podem causar nas redes *WLAN*. Sendo assim, será necessário testar este ambiente com dezenas de usuários conectados ao jogo. De posse do comportamento dos clientes e do servidor, pode-se implementar um algoritmo que se comporta exatamente como o jogo no ambiente real, com uma grande vantagem, porque pode-se definir o número de elementos nesta rede. Cenários com diversos servidores e clientes interagindo na rede são possíveis através de programas de simulação. Um ponto importante é que os resultados dos experimentos simulados são validados pelos resultados dos experimentos em laboratório.

O conteúdo desta dissertação está organizado da seguinte forma: O capítulo 2 apresenta os conceitos necessários para o entendimento deste trabalho. Nesse capítulo serão apresentados conceitos referentes a jogos de computadores, redes sem fio, simulações assim como trabalhos relacionados. No capítulo 3, é apresentada uma descrição da metodologia adotada, os cenários onde foram coletadas as informações e o modelo proposto. O capítulo 4 apresenta os experimentos reais de forma detalhada e os seus resultados. O capítulo 5 apresenta as simulações e os resultados das simulações. O capítulo 6 apresenta as conclusões da dissertação e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Este capítulo irá apresentar conceitos importantes, tanto no que se refere a jogos eletrônicos, rede sem fio e simulações em ambientes computacionais.

2.1 Jogos

A história do jogo se mistura a história do homem, mesmo antes do advento da computação nós já jogávamos, mas foi no século passado, por volta dos anos 60, que os jogos começaram a permear os meios eletrônicos. E com o desenvolvimento de máquinas de jogos eletrônicos nos anos 70 e 80, jovens aventureiros estavam criando uma nova forma de interação entre homem e máquina. Sempre existiu a polêmica sobre o desenvolvimento do primeiro jogo eletrônico, porém em (RUTTER, 2006) é citada a seguinte ordem cronológica. Em 1952 Alexander Douglas estudante de *PhD* da universidade de Cambridge produziu uma versão para o jogo *tic-tac-toe* (jogo da velha), em 1958 utilizando um osciloscópio, um físico do *BrookHaven National Laboratory* criou uma versão simplificada para um simulador de tênis, e em 1962 foi criado o famoso *spacewars* por um time de pesquisadores do MIT (*Massachusetts Institute of Technology*).

Inicialmente este novo mercado era composto por máquinas de fliperamas (máquina eletro-mecânica), onde os usuários inseriam uma moeda na máquina, o que lhes davam acesso ao jogo. O usuário jogava e ao perder ou terminar o seu tempo a jogada era finalizada. Depois vieram os videogames. A idéia era simples, porque não oferecer aos usuários os mesmos jogos em um equipamento menor que poderia ser conectado ao seu televisor. Após comprar o console, o usuário poderia comprar os cartuchos com o jogo e assim teríamos uma nova forma de vender e consumir jogos. O *Magnavox Odyssey* foi o primeiro videogame comercializado e vendeu cerca de 100.000 unidades (RUTTER, 2006).

Mas o grande momento desta indústria chegou junto com o advento do *PC* (*personal computer*). Neste momento a indústria do entretenimento digital viu o grande potencial deste mercado e uma infinidade de jogos dos mais variados gêneros, foram pro-

duzidos para computadores. Inicialmente os jogos eram simples, sua jogabilidade deixava a desejar, eram jogos com poucos recursos visuais e sonoros. À medida que os computadores foram evoluindo, os jogos também evoluíram, no que se refere à parte gráfica e outros aspectos, como por exemplo: som, física, inteligência artificial e interatividade. As figuras 1 e 2 exibem dois jogos clássicos, mas em momentos diferentes. Nota-se por esta comparação que houve um grande avanço na qualidade gráfica dos jogos, mas outros aspectos também evoluíram. No começo os jogos utilizavam de arquivos de som no formato *MIDI* (*musical instrument digital interface*), hoje em dia existem diversos formatos que podem ser utilizados em jogos. Para se produzir um jogo de sucesso, atualmente contrata-se uma orquestra para tocar as músicas que fazem parte da trilha sonora do jogo. A inteligência artificial (*IA*) dos jogos também evoluiu, nos primórdios seus oponentes no jogo executavam operações pré programadas e com pouca prática no jogo era possível identificar estas rotinas pré-estabelecidas. Atualmente, a *IA* está bem mais evoluída e os oponentes no jogo podem em alguns casos, aprender e tomar decisões coordenadas (MILLINGTON, 2006).

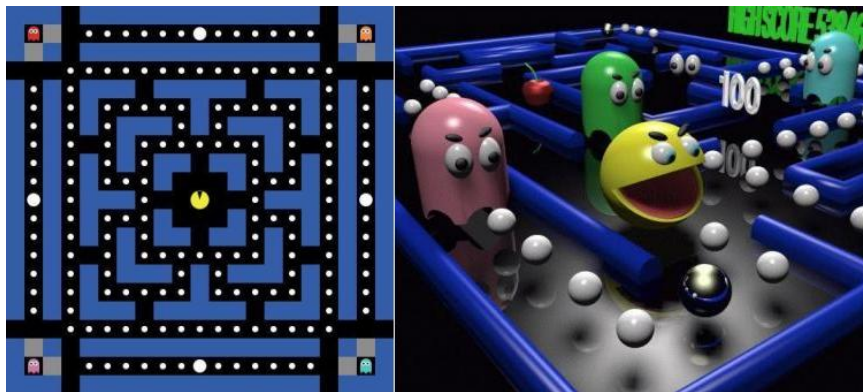


Figura 1: Comparação entre dois clássicos do videogame - PAGMAN

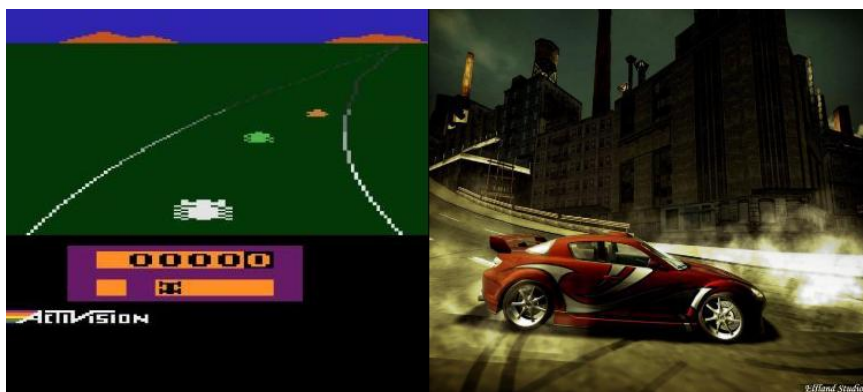


Figura 2: Comparação entre dois clássicos do videogame - ENDURO

Existem diversas classificações para os jogos eletrônicos, a classificação apresentada nesse trabalho é uma síntese da classificação apresentada em (AZEVEDO, 2005). Pode-se notar que o *FPS* aqui é descrito como um gênero, na verdade o que houve foi que este tipo de jogo se tornou tão popular que vários desenvolvedores criaram jogos com as características dos *FPS*, criando assim um novo gênero.

- **FPS** - Jogos de tiro em primeira pessoa (*First Person Shooters*)

Neste estilo de jogo, o usuário não visualiza seu personagem, ele vê somente a mão do personagem. Desta forma ele tem a impressão de que ele é o próprio personagem. Daí a idéia do estilo, tiro em primeira pessoa (EU). A arquitetura deste jogo é cliente/servidor. O usuário instala um cliente em seu computador e se conecta a um servidor na internet ou em uma rede local (*LAN*). Existem várias fases e mundos em que o usuário pode se conectar, uma vez dentro do jogo o usuário tem como objetivo principal eliminar o maior número de oponentes por partida. As partidas são em geral configuradas para durar 20 minutos.

- **RTS** - Jogos de estratégia em tempo real (*Real Time Strategy*)

Neste gênero de jogo, o principal objetivo do jogador é o de dominar parte do mapa, construir e evoluir cidades e fortalezas. E a partir de determinado ponto, ter forças para atacar seus oponentes. Para isso ele precisará recolher recursos, como por exemplo: madeira, minerais, alimentação e outros.

- **RPG** (*Role Playing Game*)

É um gênero de jogo em que o usuário irá interagir com vários outros jogadores, em um mundo, continente ou planeta desconhecido. Em geral, o objetivo do jogar é explorar este novo mundo. Ele deve buscar recursos neste mundo, como por exemplo: minerais, pedras preciosas e alimento. Ele precisa destes recursos para evoluir seu personagem, tornando-o cada vez mais forte. Em um cenário de jogo *RPG* pode-se ter dezenas ou centenas de usuários conectados.

- **MMORPG** - Jogos de *multiplayer* massivos RPG (*Massively Multiplayer Online Role Playing Game*)

Este gênero é similar ao *RPG*. A grande diferença entre o *RPG* e os *MMORPG* é a quantidade de jogadores conectados. Enquanto se tem dezenas e até centenas de pessoas conectadas a um servidor de *RPG*, pode-se ter milhares de pessoas conectadas ao *MMORPG*.

- **NRT** - Jogos de tempo não real (*Non Real Time*)

São jogos que não possuem uma interação direta, pois um jogador pode realizar uma ação e aguardar por horas ou dias que o seu oponente realize a sua ação. Um exemplo de um jogo que pode ser *NRT* é o xadrez.

A tabela 1 descreve de maneira detalhadas as características destes quatro gêneros, suas especificações e arquitetura.

	FPS	RTS	MMORPG	NRT
Arquitetura	Cliente/Servidor	Cliente/Servidor, P2P	Cliente/Servidor	P2P
Jogadores	Poucos	Poucos	Muitos	-
Servidores	Muitos	Poucos	Poucos	Nenhum
Persistência	Curta	Curta	Longa	-
Exemplos	Quake, Half Life	Starcraft, Warcraft	WoW, Ultima	Xadrez

Tabela 1: Especificações dos jogos segundo seu gênero.

Na tabela 1, observa-se que cada gênero possui suas definições de arquitetura, capacidade de jogadores conectados e etc. Como exemplo, O gênero *FPS* pode utilizar uma arquitetura cliente/servidor ou uma arquitetura *P2P*. Ou seja, pode-se configurar um servidor para ser o elemento centralizador do jogo ou pode-se utilizar a arquitetura *P2P*. Nesta última não temos a definição de um servidor. O item persistência indica quanto tempo o jogador fica conectado ao jogo. Em geral partidas em jogos *FPS* têm duração de 20 minutos, mas em jogos *MMORPG* os jogadores podem ficar conectados por horas, dias e até mesmo meses. E por fim, o item jogadores indica quantos jogadores podem estar conectados ao jogo. Em jogos *FPS* dezenas de pessoas podem jogar simultaneamente, já para os jogos *MMORPG* centenas ou milhares de pessoas podem estar conectadas.

O jogo selecionado para realizar os experimentos reais foi o *OpenArena*. O *OpenArena* foi desenvolvido utilizando ferramentas gratuitas e de código aberto. A engine¹ do *OpenArena* se baseia na *ioquake3* (engine utilizada para desenvolver o Quake3). A engine *ioquake3* é construída sobre o código fonte do Quake 3 que foi liberado pela *Id Software*² em agosto de 2005. O objetivo do projeto é criar uma distribuição *open source* do Quake 3. O *OpenArena* atende a todos os requisitos do trabalho proposto, pois este jogo pertence a classe de jogos de *FPS*, o comportamento do tráfego de rede é o mesmo dos jogos *FPS*, ou seja, o tráfego é em rajadas.

¹Traduzido: Motor do jogo - É um programa de computador ou um conjunto de bibliotecas utilizadas para se desenvolver jogos

²Uma empresa do Texas USA, que trabalha com desenvolvimento de jogos de computador

O objetivo principal do jogo é marcar pontos eliminando os adversários, usando um conjunto de armas disponíveis em diferentes ambientes. Os tipos de modalidade de jogos existentes no *OpenArena* são definidos em (OPENARENA-WIKI, 2009).

Não é necessário ter um computador muito potente para executar este jogo. Os requisitos mínimos exigidos são baixos. Possuir um processador *Intel Pentium II* 233 MHZ, 64 MB de memória RAM, uma placa de vídeo com 16MB de memória com suporte *OpenGL* e 300 MB de espaço em disco. Atualmente os *notebooks* padrão de mercado já possuem uma configuração bem superior a descrita.

O jogo contém três cenários de conflito:

- **Deathmatch** - Nesse modo de jogo o objetivo é eliminar o maior número de adversários, o jogo finaliza normalmente quando um número limite de *frags* (mortes) é atingido ou o tempo máximo é esgotado (20 minutos).
- **Team Deathmatch** - O objetivo é semelhante ao *deathmatch*, mas os jogadores são divididos em equipes, a equipe vencedora é a que obtiver o maior número de *frags*.
- **Tournament** - Torneio - Nesta modalidade apenas dois jogadores estão interagindo, os demais permanecem como espectadores do jogo. A partida termina quando um jogador é eliminado. O vencedor fica no jogo para desafiar o próximo adversário.
- **Capture the flag (CTF)** - Rouba bandeira - Os jogadores são divididos em equipes e o objetivo principal é capturar a bandeira do adversário, o jogo acaba quando o limite máximo de capturas é alcançado ou o tempo máximo é esgotado.

O *OpenArena* é um jogo gratuito e de código aberto. Sendo assim, este jogo pode ser baixado através do *link* (OPENARENA-FTP, 2009). A grande vantagem na utilização deste jogo é que pode-se realizar a pesquisa com qualquer quantidade de clientes, sem a necessidade de adquirir licenças do *software* para cada estação. O jogo é multiplataforma, podendo ser executado em estações com *Microsoft Windows*, *Linux*, *Mac* e *OpenBSD*. Ser de código fonte aberto, fato de grande importância, pois em alguns casos o pesquisador poderá ter acesso ao núcleo do jogo, para coletar informações referentes ao comportamento ou mesmo propor implementações que melhorem o comportamento do jogo, no que se refere à conectividade e utilização dos recursos de rede.

2.2 Rede Wireless

Existem várias formas de conexões possíveis de serem implementadas, todas possibilitando a conexão entre os computadores conectados a ela. Neste trabalho utilizou-se as redes *WLAN*, nesta rede seus elementos se comunicam através de ondas de rádio frequência. O padrão mais difundido no mercado atualmente é o 802.11g, com esta tecnologia o usuário pode se comunicar com outras máquinas a uma velocidade nominal de até 54 Mbps. Para realizar os cálculos de utilização da rede e construção da simulação, utilizamos a taxa efetiva da rede. Apesar da taxa nominal da rede *Wi-Fi* 802.11g ser de 54 Mbps sabe-se que a taxa efetiva deste padrão está em torno de 24 Mbps até 36 Mbps (LV SHAOHE; WANG, 2008). Para o equipamento que utilizamos, testes mostraram que sua taxa efetiva é de 21.7 Mbps (INFO-REVIEWS, 2010).

A seguir, serão apresentados os conceitos utilizados para realizar as simulações deste trabalho e por fim uma breve abordagem sobre o *NS2* (*network simulation*).

No capítulo 4 é descrito detalhadamente os experimentos executados e suas fases, esta parte do trabalho é importante, pois de posse dos dados coletados nesses experimentos foram extraídos os dados que foram utilizados como parâmetros nas simulações.

O *Network Simulation*, mais conhecido como *NS2* é um simulador de eventos discretos em redes. A versão utilizada foi a *NS-2.34*. O *NS2* é muito popular no meio acadêmico e isso se deve ao fato de ser uma aplicação de código fonte aberto (*GNU General Public License*), implementar diversos tipos de redes e protocolos. O *NS2* foi criado pela universidade de Berkeley. Para se criar *scripts* de simulação no *NS2*, o pesquisador precisa conhecer as duas linguagens utilizadas pelo simulador, são elas: *C++* e *Tcl*. Os *scripts* em *Tcl* são interpretados pelo simulador, já os objetos utilizados nas simulações são implementados em *C++* (NS2, 2009; NS2-WIKI, 2009).

Um modelo poderá mostrar como esses jogos irão se comportar ao longo do tempo, qual será o comportamento quando um determinado número de jogadores estiver interagindo. Métricas que verificam o tempo de resposta da aplicação são utilizadas para diagnosticar o comportamento do jogo.

(FARBER, 2002) descreve o comportamento do tráfego do jogo. O tráfego do servidor do jogo acontece em rajadas. Os clientes do jogo mandam dados para o servidor a todo instante, o servidor por sua vez analisa esses dados, realiza um agrupamento dos dados e manda uma rajada de mensagens para todos os clientes, assim os clientes podem atualizar seu cenário com as informações dos seus adversários. Foram realizadas coletas

de dados em dois gêneros de jogos distintos, o *FPS* e o gênero de estratégia. O tempo limite aceitável entre o envio e o recebimento de dados nos jogos *FPS* deve ser menor que 150 ms. A partir deste ponto, a jogabilidade é prejudicada. Já para jogos de estratégia, este limite fica em torno de 500 ms.

(FARBER, 2002) conclui que a discussão sobre *QoE* em jogos está longe de acabar, mas que é necessário a construção de modelos para se entender o comportamento dos jogos.

Em (LANG TANJA; BRANCH, 2004) é proposto um modelo sintético para avaliar o comportamento do *Quake 3*. Nas análises realizadas 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 irá variar com relação ao número de jogadores conectados ao jogo.

(LANG TANJA; BRANCH, 2004) conclui que é importante a construção de modelos que possam auxiliar desenvolvedores de jogos e provedores de serviço no que se refere ao comportamento do tráfego de jogos em redes.

(CLAYPOOL, 2005) realiza um estudo e faz comparações entre os consoles *Nintendo DS* e o *Sony PSP Hand-held* em uma rede sem fio IEEE 802.11. Comparando o comportamento do tráfego de vários gêneros de jogos, realiza a caracterização de varias fases envolvidas no processo do jogo, como por exemplo: a fase sincronização, do jogo propriamente dito e da mudança de cenários.

(CLAYPOOL, 2005) diz que o crescimento do mercado de jogos digitais e das redes *wireless* faz com que o mercado ofereça novas formas de interação entre jogadores.

Em (ZANDER, 2005) foi desenvolvido um modelo de tráfego para analisar o comportamento do *Halo 2* para *Xbox*. O modelo apresentado destina-se a analisar aspectos como largura de banda consumida, quantidade de pacotes na rede e o intervalo entre recebimentos de pacotes. (ZANDER, 2005) conclui que o seu modelo será útil para desenvolvedores e provedores de acesso no que se refere ao tráfego gerado pelo *Halo 2* nas redes. De posse destas informações, provedores e desenvolvedores poderão verificar o impacto que o jogo causa na rede.

Em (BRANCH PHILIP; ARMITAGE, 2006), foram analisados jogos de tiro em primeira pessoa (*FPS*). Foi proposta uma técnica para se modelar a distribuição do tamanho dos

pacotes do jogo em decorrência ao aumento no número de jogadores conectados. Para isso, foram analisados o comportamento de seis importantes jogos do gênero *FPS*. (BRANCH PHILIP; ARMITAGE, 2006) conclui que é fundamental que se conheça esta distribuição para se criar modelos de simulação mais realísticos.

Em (BRUN JEREMY; SAFAEI, 2006) é proposto um modelo para se escolher o melhor servidor na rede. Tendo em vista que esta escolha irá impactar diretamente na jogabilidade, foi criada a noção de “tempo crítico de resposta” e as escolhas baseadas no modelo tem impacto direto na jogabilidade. O modelo apresentado em (BRUN JEREMY; SAFAEI, 2006) funcionou bem para redes pequenas. Para redes com um número maior de elementos foi utiliza heurísticas.

Em (OLIVER IAIN; MILLER, 2010) é descrito o comportamento de jogos de simulação, como o *Second Life (SL)*. Nesse artigo é apresentado a largura de banda requerida por alguns jogos eletrônicos. Segundo (OLIVER IAIN; MILLER, 2010) jogos *FPS* necessitam de aproximadamente 67 Kbps de largura de banda. Em contra partida, jogos do tipo *SL* podem requerer até 775 Kbps.

Em (S RATTI; B, 2010) é realizado 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. Também é realizado um comparativo entre os trabalhos existentes na literatura que tratam deste assunto, conforme é apresentado, alguns parâmetros podem ter alguma divergência, isso se deve ao tipo de análise que foi realizado. Por fim (S RATTI; B, 2010) sugere que diferentes jogos do gênero *FPS* têm comportamento semelhante e que os dados analisados se aplicam a um variedade maior de jogos deste gênero.

3 DESCRIÇÃO DA METODOLOGIA E DO MODELO PROPOSTO

Conforme mencionado, este trabalho engloba três etapas:

1. Coleta de dados reais
2. Desenvolvimento de simulações dos cenários de teste e validação das simulações
3. Avaliação de outros cenários, através das simulações

A seguir, a metodologia de cada etapa será descrita com seus procedimentos detalhados.

Para os experimentos e coleta de dados foram utilizados diversos computadores (*notebooks*) conectados a uma rede *wireless* configurada especialmente para esse fim (ambiente controlado). Para o experimento 1 a rede *WLAN* foi instalada no laboratório do mestrado em informática da PUC-Minas São Gabriel e para o experimento 2, a rede *WLAN* foi montada no *MIC* (*centro de inovação tecnológica da Microsoft*) localizado na mesma unidade da PUC-Minas. A rede foi composta por um *Access Point (AP) Linksys WRT54G* e vários *notebooks* conectados ao *AP*. Esses equipamentos utilizam a tecnologia 802.11 b/g que proporciona desempenho nominal teórico de até 54 Mbps e prático de 21.7 Mbps. O *AP* foi configurado para permitir o acesso apenas dos membros (voluntários) dos experimentos. Dessa forma, somente os voluntários do projeto estavam conectados à rede *WLAN* dos experimentos. Para o experimento 2, utilizamos o *link* de internet do *MIC*, que conta com um acesso a internet através de um *link* veloz¹ de 2 Mbps. Essa rede destina-se ao acesso de alunos e professores. Como os experimentos foram realizados no sábado, não havia outros usuários utilizando esse *link*. Então, apenas os membros do projeto que estavam conectados à *WLAN* poderiam utilizar o *link* de internet.

Utilizando o ambiente controlado, foi criado um ambiente de testes. Nesse ambiente, um *notebook* estava executando a aplicação do jogo como servidor e os outros

¹Produto oferecido por uma operadora de telefonia para prover acesso à internet.

notebooks configurados como clientes do jogo. Nesse ambiente de teste os jogadores interagiram entre si, jogando uma série de partidas do *OpenArena*. Os dados gerados pelo servidor e pelos clientes foram capturados através de uma ferramenta que armazenou os dados que estavam trafegando na rede. Inicialmente, a única aplicação rodando na rede era a do jogo e, posteriormente, foram acrescentadas outras aplicações. Sendo assim, foram feitos dois tipos de experimentos. No experimento 1, apenas o jogo estava sendo executado e, no experimento 2, havia o jogo e outras aplicações sendo executadas na rede. Na Seção 3.2, serão explicados os detalhes relacionados aos experimentos.

3.1 Ferramenta de captura de dados

A ferramenta selecionada para realizar a captura dos dados que trafegaram na rede foi a *wireshark* em sua versão 1.0.6. Esta ferramenta foi escolhida por ser multi-plataforma, podendo ser executada no *Microsoft Windows XP*, *Microsoft Windows Vista*, *OS-X*, *Unix* e *Linux*, ser gratuita e possuir suporte a centenas de protocolos (WIRESHARK, 2009).

A ferramenta foi instalada no servidor e no *notebooks* que estavam executando o cliente do jogo. Foi monitorado todo o tráfego gerado pelo servidor e pelos clientes. Todos os dados que estavam trafegando na rede *WLAN* foram armazenados em arquivos para posterior análise. No experimento 2, quando havia outras aplicações sendo executadas, também foi preciso instalar a ferramenta em cada um dos *notebooks*. Sendo assim, os *notebooks* que estavam acessando a internet, seja para navegação interativa ou acessando vídeos, tinham a ferramenta instalada. Dessa forma, foi capturado e monitorado o comportamento de cada estação no ambiente controlado.

3.2 Descrição do ambiente de testes

Para os dois experimentos foi criado um servidor com as seguintes configurações: *notebook* HP Pavilion DV2000, com 4 GB RAM, 160 HD e processador AMD Turion 64Bits 1.8 GHz, placa de rede compatível 802.11 b/g. Os *notebooks* que rodaram a aplicação em modo cliente possuíam como característica principal uma placa de rede *wireless* padrão 802.11b/g. Esses *notebooks* possuíam configurações compatíveis com as exigências para se rodar o jogo selecionado.

A cada rodada de teste, o número de jogadores era gradualmente incrementado, conforme mostra a tabela 2. Assim, foi possível verificar o comportamento do ambiente

de simulação em relação ao aumento de jogadores conectados.

De posse das informações coletadas, avaliou-se o impacto que o tráfego gerado pelo jogo causou nas demais aplicações executadas e vice-versa. Os dados coletados também foram utilizados na construção das simulações, pois estes dados foram transformados em parâmetros de entrada para o simulador.

Para se obter maior fidelidade nos experimentos reais, existiam outras aplicações, no experimento 2, concorrendo com o jogo. O que se desejou no experimento 2 foi aproximar os experimentos o máximo possível do que ocorre em um ambiente real, como por exemplo, em uma *LAN house*.

A coleta de dados levou em consideração dois momentos, chamados de experimento 1 e experimento 2. No primeiro, o jogo era a única aplicação que estava rodando na rede. No segundo, além do jogo, havia outras aplicações concorrentes. A coluna **Outras Aplicações** das tabelas 2 e 3 indicam se o jogo estava ou não concorrendo com outra aplicação na rede. Sendo assim, os experimentos foram executados várias vezes, em dias diferentes, para se ter uma idéia geral do comportamento da rede *WLAN*.

Fases	Servidor	Clientes	Tempo de coleta	Com Outras Aplicações
1	1	1	20 Minutos	Não
2	1	2	20 Minutos	Não
3	1	3	20 Minutos	Não
4	1	4	20 Minutos	Não
5	1	5	20 Minutos	Não
6	1	6	20 Minutos	Não

Tabela 2: Cenário para coleta de dados - Experimento 1 (Experimentos reais).

A tabela 2 representa as fases do experimento 1. Pode-se verificar que, no experimento 1, apenas o jogo estava sendo executado na rede. Sabe-se que usuários de redes *wireless*, em geral, utilizam a rede para navegar na internet, assistir a vídeos, baixar arquivos e conversar com outras pessoas através de *chat* utilizando o navegador ou mesmo programas de mensagens instantâneas. Então, essas ações também devem ser executadas, para que se possa analisar o tráfego gerado por esses cenários mistos. A tabela 3 apresenta o comportamento do ambiente de teste. Nela, observa-se que as fases do experimento 2 são compostas por um servidor, seis clientes e outros usuários, conectados ao ambiente, executando outras aplicações.

Cada fase de cada experimento teve duração de vinte minutos. O número de *notebooks* entre as fases do experimento 1 sempre é incrementado de um em um. Então,

Fases	Servidor	Clientes	Tempo de coleta	Com Outras Aplicações
1	1	6	20 Minutos	Sim
2	1	6	20 Minutos	Sim
3	1	6	20 Minutos	Sim

Tabela 3: Cenário para coleta de dados - Experimento 2 (Experimentos reais).

no experimento 1 temos os cenários que são apresentados na tabela 4.

Fase do Experimento 1	Equipamento	Tempo de Coleta
1	1 Cliente + 1 Servidor	20 Minutos
2	2 Cliente + 1 Servidor	20 Minutos
3	3 Cliente + 1 Servidor	20 Minutos
4	4 Cliente + 1 Servidor	20 Minutos
5	5 Cliente + 1 Servidor	20 Minutos
6	6 Cliente + 1 Servidor	20 Minutos

Tabela 4: Detalhamento das fases dos experimentos 1 (Somente o jogo sendo executado).

O detalhamento da tabela 4 exhibe as fases do experimento 1, onde pode-se observar que apenas o jogo está sendo executado, com incremento de um até sete *notebooks* conectados ao ambiente. Um estava executando o servidor do jogo e os outros seis foram conectados como clientes do jogo.

A tabela 5 apresenta o detalhamento das fases do experimento 2. Esse experimento foi dividido em três fases. Em cada uma delas, tínhamos, além do jogo, outras aplicações sendo executadas. As fases compostas pelo último cenário do experimento 1, ou seja, um *notebook* como servidor e seis *notebooks* como clientes do jogo e outras aplicações. Essas aplicações foram divididas da seguinte forma:

- O jogo (Servidor + Clientes)
- Navegação interativa
- Acesso a vídeos *online*
- Troca de arquivos entre *notebooks* conectados ao ambiente de simulação

Analisando a tabela 5, observa-se o comportamento geral dos participantes ao longo do experimento 2. Na fase 1, havia sete *notebooks* executando o jogo, um como servidor e seis como clientes do jogo, outro *notebook* navegando na *WEB* e outro conectado a um

Fase do Experimento 2	Tipo de Aplicação
1	Jogo + 1 Navegação Web + 1 Vídeo
2	Jogo + 2 Navegação Web + 1 Vídeo
3	Jogo + 1 Navegação Web + Troca de Arquivos

Tabela 5: Detalhamento das fases dos experimentos com outras aplicações (Experimento 2).

site de *streaming* de vídeo. Para a fase 2, foi acrescentado mais um *notebook* navegando na *WEB*. Por fim, na fase 3 temos além do jogo e navegação na *WEB* dois *notebooks* trocando arquivos via FTP.

A figura 3 apresenta o esboço do ambiente de simulação para a coleta de dados. Nessa figura, é representada a fase 6 do experimento 1, onde havia um servidor e seis estações cliente executando o jogo.

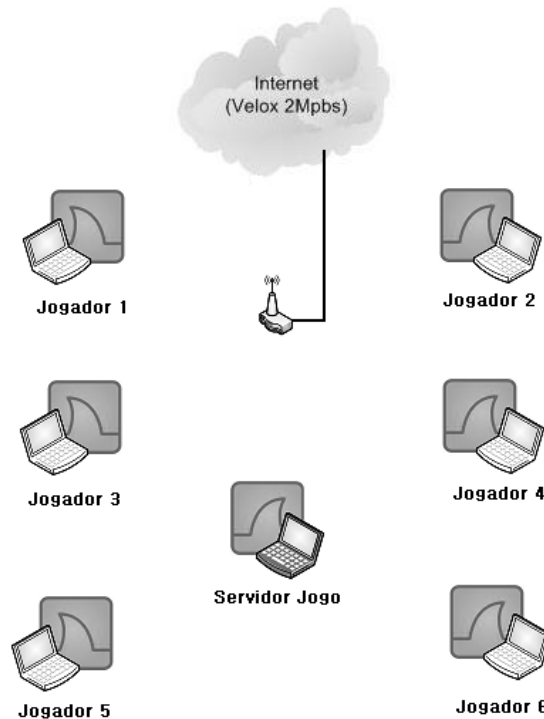


Figura 3: Experimento 1. Um servidor e seis clientes.

Os dois tipos de experimentos seguiram a mesma estrutura. O que variou entre as fases dos experimentos foi o número de clientes conectados, conforme mostram as tabelas 4 e 5. O tempo de coleta das informações foi de 20 minutos.

No primeiro experimento, representado pela figura 3, o cenário é formado pelos seguintes elementos:

- Um *link* de internet de 2 Mbps;
- Seis *notebooks* conectados a um roteador *wireless*, executado a aplicação do jogo em modo cliente;
- Um *notebook* executando a aplicação do jogo em modo servidor;
- Um roteador *wireless Linksys WRT54*;

Abaixo são apresentadas algumas especificações e configurações do roteador *Linksys WRT54*. Para os experimentos, as *features*² do roteador foram configurados da seguinte forma:

- SSID: Mestrado
- Segurança: WPA2
- MAC: 802.11g
- Servidor *DHCP*: Habilitado;
- Canal: 6

Com essas configurações, não foram necessárias configurações adicionais nos *notebooks* dos colaboradores para a entrada na rede. A segurança foi habilitada para se ter maior controle sobre os usuários conectados a rede *WLAN*, evitando, assim, possíveis interferências nos experimentos. O servidor *DHCP* do roteador foi habilitado para facilitar a entrada do usuário na rede sem a necessidade de maiores configurações.

3.3 Modelo Proposto

Conforme dito anteriormente, este trabalho propõe a criação de um modelo de tráfego de um jogo do tipo *FPS* em uma rede sem fio. Para se criar o modelo, foi necessária a criação do ambiente controlado, a fim de identificar os parâmetros de entrada e saída

²Opções que compõe as funcionalidades de um equipamento.

da rede. Após a realização de diversas coletas, os dados foram analisados e selecionados como base de entrada para a criação do modelo de simulações.

Utilizando o **NS2** (NS2, 2009), foi construído o modelo de simulação. A idéia foi utilizar o simulador para realizar as mesmas interações que ocorrem em um ambiente real. Assim, verificou-se, inicialmente, a validade do modelo proposto, através de comparações de resultados das simulações e das coletas. Com isso, o modelo pode ser usado para cenários diferentes. Os dados que foram coletados durante a execução das partidas reais foram utilizados na configuração. Por exemplo, o tempo de intervalo entre o envio de pacotes na rede, o tamanho médio dos pacotes para os clientes e o servidor entre outros, foram parâmetros de entrada da simulação, obtidos através das coletas. O modelo proposto foi validado pelo comportamento real, capturado na etapa de coleta de dados deste trabalho. Comparações dos dados de saída foram feitas e, após a validação do modelo, o número de *nós* na rede foi alterado. O objetivo foi verificar o comportamento da rede quando o número de elementos na rede aumenta e mensurar qual é o desempenho quando há vários usuários conectados à rede, tanto para os que executam o jogo, quanto para os que executam outras aplicações.

A Figura 4, esboça o modelo de simulação criado. Nela, é apresentada, de forma resumida, como se deu a passagem dos dados observados nas coletas para o simulador. Essa figura ilustra um cenário do experimento 1. O mesmo princípio foi utilizado em todas as simulações realizadas. Nas simulações, o *nó* 0 (zero) sempre representa o *AP*. O *nó* um sempre representa o servidor do jogo e os demais *nós* são os clientes do jogo. Pode-se observar que, próximo a cada *nó*, há um retângulo com a descrição do *nó* e seus parâmetros de entrada. Após a execução da simulação, os dados de saída são armazenados em um arquivo de *trace*. Esse evento está representado pela linha que liga cada *nó* ao círculo com a legenda “Dados de Saída da Simulação”, na figura.

Como, já mencionamos, o objetivo do modelo de simulação foi à extrapolação do número de elementos na rede. Os experimentos reais foram executados com um número limitado de participantes. No ambiente de simulação, este número pode ser facilmente alterado. Assim pode-se analisar o comportamento da rede com um maior número de usuários conectados. Podendo-se avaliar o impacto de tráfego do jogo em cenários que refletem o mundo real das redes *Wi-Fi* de *shoppings*, clubes, *LAN Houses*, universidade etc.

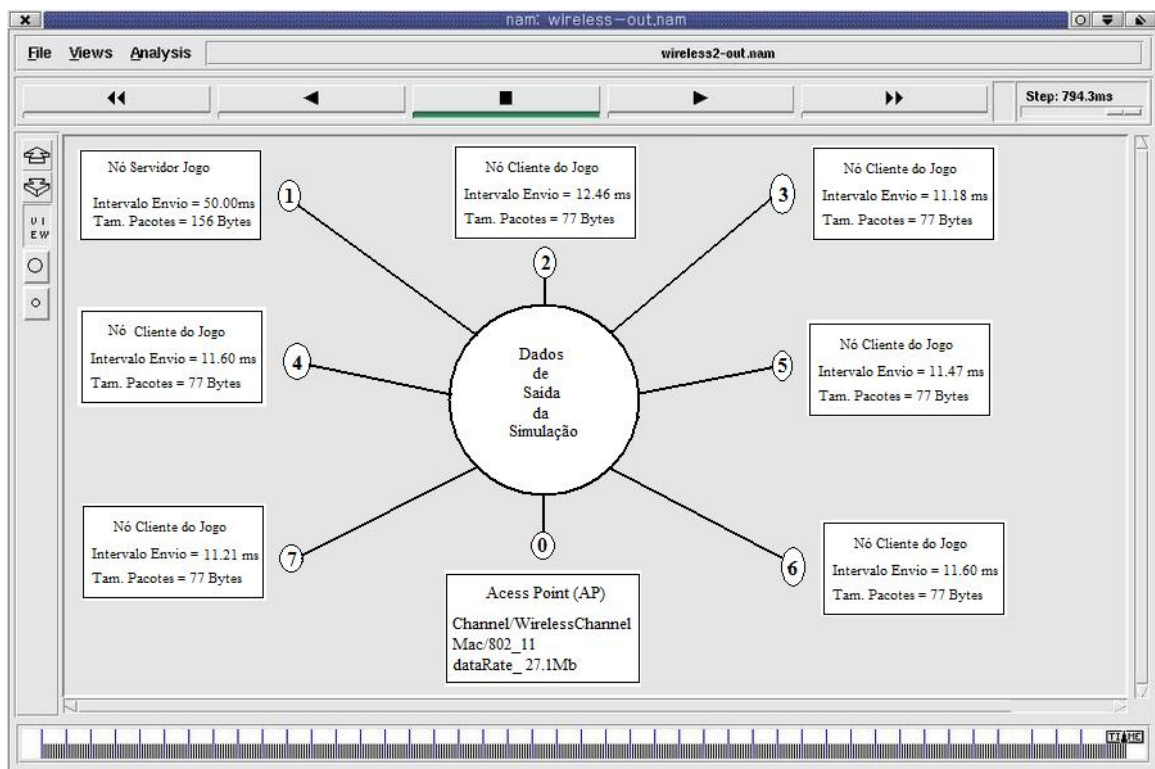


Figura 4: Esboço do modelo criado no NS2.

4 EXPERIMENTOS E RESULTADOS

Este capítulo descreve os resultados obtidos nos experimentos reais. As próximas seções apresentam os resultados das coletas e as análises de resultados dos experimentos 1 e 2. Conforme dito anteriormente, os experimentos contaram com a participação de voluntários, cada um utilizando um *notebook* com placa *wireless* compatível com o padrão 802.11g. Foram realizadas as configurações de segurança no roteador. Assim, somente os integrantes do projeto puderam ter acesso à rede. Evitou-se possíveis interferências ou concorrência de tráfego. As características dos *notebooks* dos voluntários obedeceram às exigências mínimas para se executar o jogo, mencionadas no capítulo 2.

O jogo selecionado para os experimentos foi o *OpenArena*, e a ferramenta de captura foi o *software wireshark*, conforme já mencionado. Foram desconsiderados os pontos de aquecimento e desaquecimento do tráfego dos dados, ou seja, as coletas dos dados refletem o comportamento dos jogadores efetivamente conectados ao ambiente do jogo.

4.1 Experimento 1

Cada fase do experimento 1, que só tinha tráfego do jogo, durou exatamente 20 minutos. Neste experimento, o jogo foi iniciado com um *notebook* cliente, na primeira fase. O número de *notebooks* clientes foi incrementado, em cada fase, gradativamente até seis *notebooks* clientes na fase 6.

O gráfico da Figura 5 apresenta o comportamento do tráfego do jogo na rede. Antes de apresentar os gráficos, faz-se necessário reforçar que, para os experimentos, são necessários no mínimo dois *notebooks*: um executando o jogo como servidor e outro como cliente do servidor do jogo. O servidor também é um jogador, mas este não envia dados para a rede, ou seja, no servidor do jogo também há um jogador. Esse jogador pode interagir com o ambiente e com os outros jogadores, mas, ao se capturar os dados apenas deste jogador, observa-se que não são transferidos dados para a rede. Sendo assim, quando temos x jogadores conectados à rede, os dados que estão sendo transferidos são de $x-1$

jogadores, pois o jogador localizado na estação servidor não manda dados para a rede, isso ocorre porque esse jogador está não própria “estação servidor” do jogo. Sendo assim, a quantidade de jogadores foi descrita explicitamente. Com a quantidade de jogadores mais o servidor, por exemplo, por três jogadores mais um servidor, entendendo-se que quatro *notebooks* estavam conectados ao ambiente do jogo.

O gráfico da Figura 5 apresenta, para cada fase do experimento 1, a vazão média do tráfego do jogo durante os 20 minutos do experimento. Os dados apresentados são apenas do jogo. O gráfico foi gerado considerando-se o eixo x como sendo o tempo decorrido do experimento a cada 1 segundo e o eixo y representa a vazão média em *Kbps*. Pode-se notar, pelo gráfico, que, a cada fase do experimento 1, a quantidade de dados que trafegam na rede aumenta. A primeira linha do gráfico representa a vazão média em *Kbps* para a fase 1 do experimento 1, a segunda linha representa a vazão média da fase 2, e assim sucessivamente até a última linha (no topo do gráfico) que representa a vazão média para a fase 6 do experimento 1.

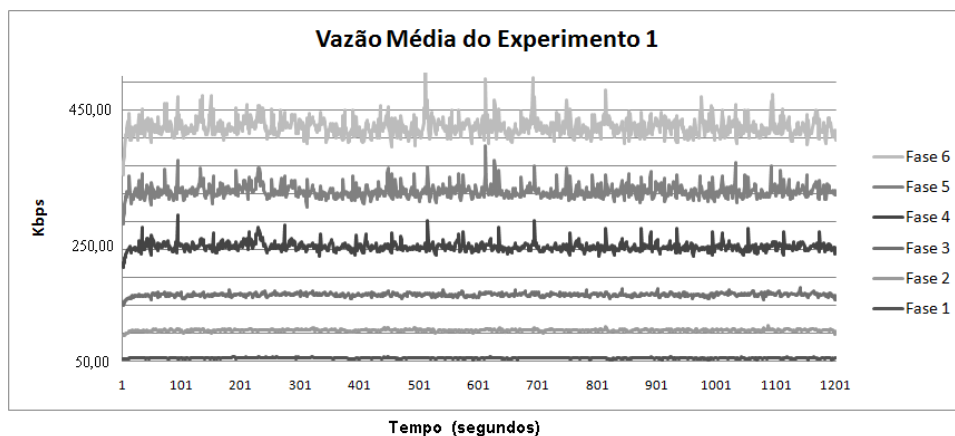


Figura 5: Vazão média do jogo para cada fase do experimento 1.

Analisando cada linha separadamente, pode-se identificar a vazão média do jogo em suas respectivas fases. Na fase 1, o *throughput*¹ médio foi 68,16 *Kbps*. Na fase 2, o *throughput* médio foi de 149,11 *Kbps*. Na fase 3, foi observado que o *throughput* foi de 190,67 *Kbps*. Na fase 4, o *throughput* foi de 257,73 *Kbps*. Na fase 5, foi de 297,08 *Kbps*, e por fim, na fase 6, o *throughput* foi de 344,70 *Kbps*. Pode-se observar que, a cada fase do experimento o valor da vazão aumenta. Isso se deve ao fato de que, a cada fase mais um jogador é inserido na rede.

A tabela 6, exibe os valores da vazão, o desvio padrão para cada fase do experimento, assim como o percentual de utilização da rede.

¹Taxa de transferência - Quantidade de dados transferido entre o servidor e os clientes.

Fases	1	2	3	4	5	6
Kbps	68,16	149,21	190,67	257,73	297,08	344,70
Desv. Padrão	2,49	4,19	6,46	21,85	25,19	32,82
Utilização/Sec	0,32%	0,69%	0,88%	1,19%	1,37%	1,59%

Tabela 6: Vazão de todas as fases dos experimentos 1.

Pode-se observar pela tabela 6, que, mesmo para a última fase do experimento 1, o percentual de ocupação da rede é baixo, apenas 1,59%. O que mostra que seis jogadores mais um servidor não causaram grandes impactos a rede, no que se refere ao consumo de banda.

O gráfico da Figura 6 apresenta o comportamento do jogo na rede em relação ao aumento de tráfego, proporcional ao crescimento no número de usuários conectados ao jogo. Esses dados foram coletados durante 20 minutos. O gráfico mostra que, a cada usuário conectado na rede, há um aumento proporcional no número de dados que estão trafegando. O crescimento na vazão do jogo só não é linear, porque, depende da capacidade de experiência do jogador, ou seja, nos experimentos observou-se que um jogador iniciante envia menos dados para a rede que um jogador experiente. Isso ocorre, porque, o jogador iniciante tende a se movimentar menos no jogo, realizando certas operações mais lentamente. Enquanto um jogador experiente tende a se movimentar mais rapidamente, e executa mais ações dentro do ambiente do jogo e, conseqüentemente, gerando mais tráfego. Outro fato que deve-se considerar é a configuração do *notebook* utilizado, pois, verificou-se nos experimentos reais que, *notebook's* com maior poder de processamento tendem a enviar mais dados para a rede.

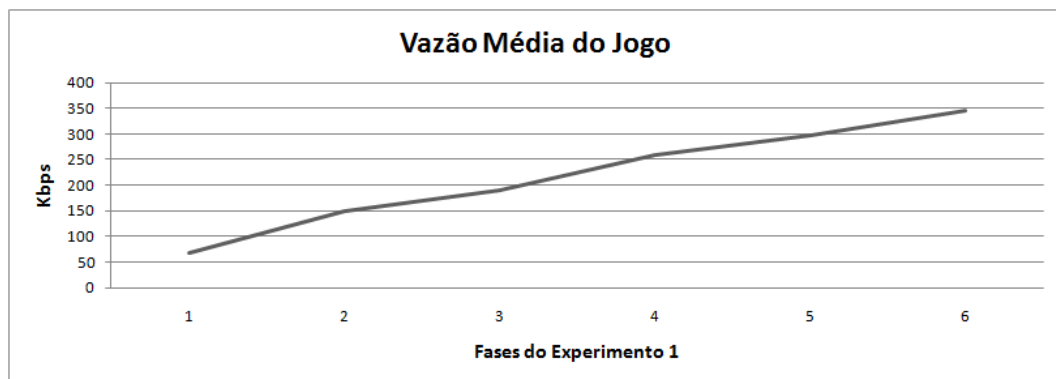


Figura 6: Vazão média das fases do experimento 1.

Uma informação importante, refere-se à quantidade de largura de banda consumida em cada fase do experimento 1. O gráfico da Figura 6, mostra em *Kbps* a quantidade de

banda consumida, à medida que os jogadores vão entrando na rede.

Observa-se no gráfico da Figura 6, a largura de banda consumida nas fases do experimento, mesmo com seis jogadores mais o servidor, não chegou a 400 Kbps da capacidade efetiva da rede que foi de 21.7 Mbps. Além desses dados, a cada cinco minutos, era verificado o tempo de resposta de cada jogador, que em nenhum momento ultrapassou os 20ms (milissegundos). Este tempo de resposta é considerado satisfatório em jogos de Tipo em Primeira Pessoa (*FPS*) segundo (FARBER, 2002).

O gráfico da Figura 7 apresenta a largura de banda consumida em percentagem. Pode-se notar nesse gráfico que, com o número máximo de jogadores conectados à rede, menos de 1,60% da largura de banda foi consumida.

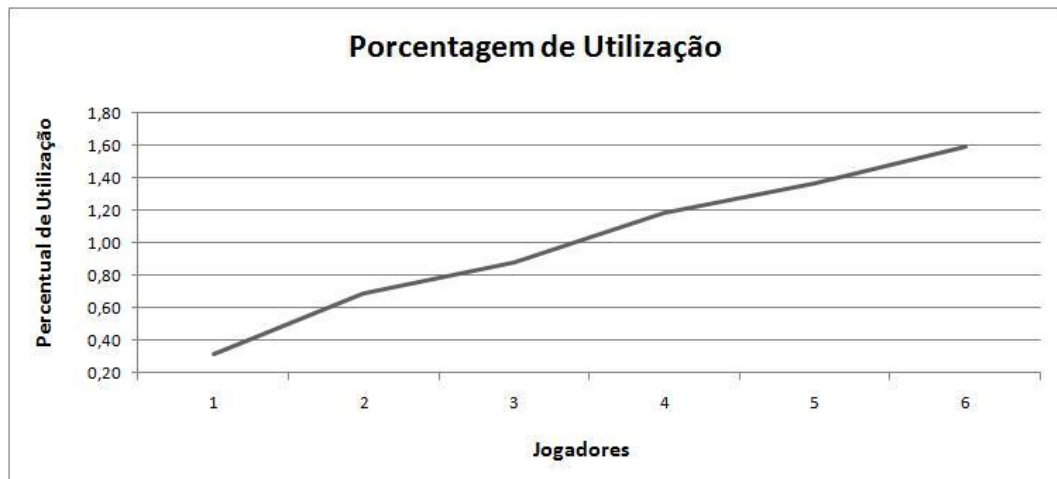


Figura 7: Jogadores VS Percentagem de utilização da rede.

4.2 Experimento 2

O experimento 2 analisou, em ambiente real, o comportamento de outras aplicações sendo executadas juntamente com o jogo. O objetivo foi o de coletar o tráfego do jogo e das outras aplicações, comparando-os.

Para realizar esse experimento, foram convidados os mesmos participantes do experimento 1. Além desses, foram convidados mais quatro voluntários para executar outras aplicações em nosso ambiente. O tráfego da rede foi coletado e analisado. Cada fase desse experimento teve duração de 20 minutos e as configurações da rede foram as mesmas utilizadas no experimento 1.

A tabela 7 mostra as fases do experimento 2 e a quantidade de usuários conectados ao ambiente controlado, em cada fase. Nota-se que, para esse experimento, o número de clientes conectados ao jogo é sempre seis. Esse experimento começa com esse número de clientes, porque foi observado nos experimentos anteriores que quantidades menores de clientes não causavam grandes problemas à rede, pois a largura da banda consumida era muito pequena. Partimos, portanto, de seis, cujo comportamento não congestiona a rede, conforme demonstrado no experimento 1.

Fase	Servidor	Clientes	Navegação Web	Vídeo	Arquivos
1	1	6	1	1	0
2	1	6	2	1	1
3	1	6	1	0	1

Tabela 7: Fases do Experimento 2 - Jogadores, servidor do jogo e outras aplicações.

A tabela 7, também mostra os tipos de aplicações que estavam concorrendo com o jogo na rede e o número de usuários executando cada aplicação, em cada fase do experimento 2. As aplicações escolhidas representam classes de *Quality of Service (QoS)* diferentes: navegação *WEB* representa a classe interativa, vídeo representa a classe *streaming* e transferência de arquivos representa a classe *background*. As classes adotadas seguem taxonomia proposta pelo grupo 3GPP (3GPP, 2009).

De acordo com a tabela 7, na fase 1 do experimento 2, havia seis *notebooks* executando o cliente do jogo, um *notebook* executando o servidor do jogo, um *notebook* navegando na *WEB* e um *notebook* acessando vídeo na internet. A fase 2 é semelhante à fase 1, com um acréscimo de mais um *notebook* navegando na *WEB*. A fase 3 contém, além das aplicações anteriores, transferência de arquivos.

Os gráficos das Figuras 8 e 9, mostram, respectivamente, a quantidade de dados trafegados e o percentual de utilização da rede para as fases 1, 2 e 3 do experimento 2.

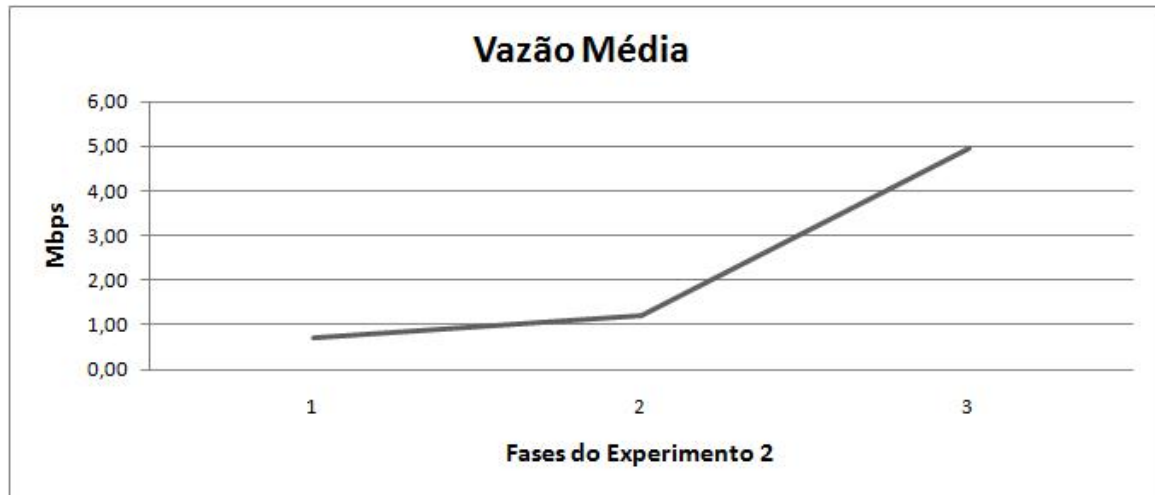


Figura 8: Vazão média das fases do experimento 2.

Pode-se verificar, pelo gráfico da Figura 8, que, a cada fase do experimento, que o consumo de banda da rede aumentou. Isso se deve ao fato de que, havia diferentes tipos de aplicações em cada fase. Além do jogo, que foi fixado com seis jogadores e um servidor, em cada fase havia um conjunto de aplicações concorrentes. Pode-se notar, na fase 1 que, o tráfego médio da rede era de 0.696 Mbps. Já para a fase 2, com a inclusão de *streaming* de vídeo este consumo subiu para 1.213 Mbps, e por fim, quando foi introduzida uma aplicação FTP o consumo subiu para quase 5 Mbps.

Esse crescimento na utilização de banda era esperado, o que não se sabia, era o comportamento e os valores destes aumentos. Estas informações são de extrema importância, pois estes dados servirão de parâmetros para as simulações e conseqüentemente irão validar ou não os *scripts* de simulação.

Pode-se observar, pelo gráfico da Figura 9, que o percentual de utilização da rede aumenta. Esse aumento é proporcional ao acréscimo de elementos da rede. Com essa observação, pode-se verificar o quanto da banda foi consumida em determinada fase do experimento. Nota-se que a fase 3 teve o maior consumo, mas este consumo não ultrapassou 22.75% da capacidade total da rede.

A tabela 8, apresenta um detalhamento dos dados capturados em cada fase do experimento 2. Conforme dito anteriormente, estes dados são importantes para a modelagem e construção do modelo de simulação. Os dados apresentados na tabela 8 são uma síntese de todo o comportamento que foi observado durante os experimentos. Ou-

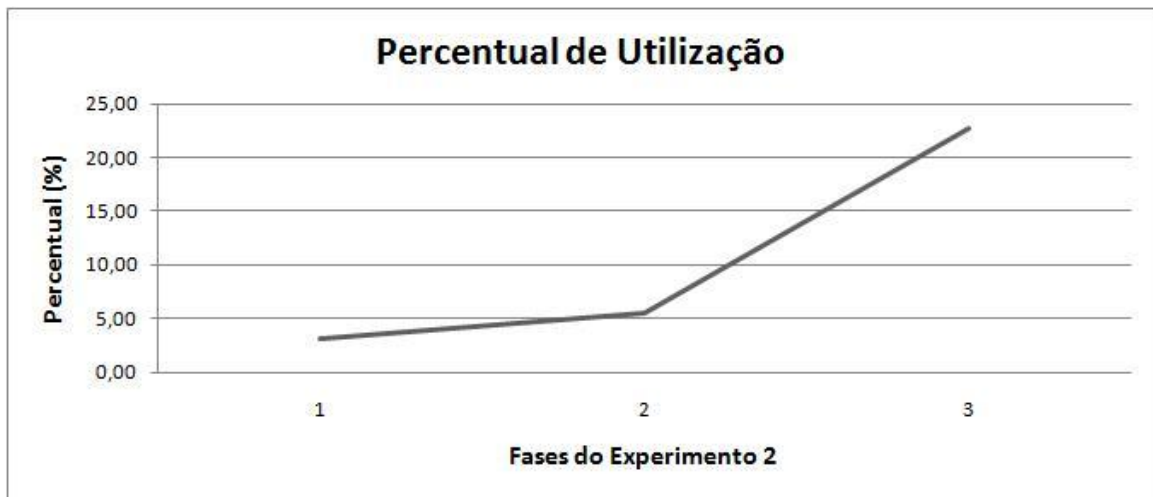


Figura 9: Percentual de utilização da rede para as fases do experimento 2.

tros parâmetros também foram extraídos das observações, como, por exemplo, tempo de resposta, quantidade de pacotes trafegado etc.

Fase	Jogo	HTTP	Vídeo	FTP	Total
Fase 1	0,370	0,083	0,243	N/A	0,696
Fase 2	0,341	0,166	0,706	N/A	1,213
Fase 3	0,386	0,706	N/A	4,479	4,936

Tabela 8: Detalhamento dos dados capturados para cada fase do experimento 2.

Observou-se nos experimentos 1 e 2 que os cenários analisados não possuíam uma quantidade suficiente de usuários conectados a rede para se perceber grandes impactos, seja na rede ou mesmo nas aplicações em execução. Notou-se claramente o aumento na quantidade de dados trafegados e no percentual de utilização da rede. Por fim, para os cenários dos experimentos 1 e 2, não foram encontrados pontos onde houvessem problemas, seja para o jogo ou para as aplicações em execução. Sendo assim, será necessário realizar simulações com um maior número de elementos na rede para se detectar possíveis cenários de gargalos. No capítulo 5 será apresentado o modelo de simulação e seus resultados.

5 MODELO DE SIMULAÇÃO E RESULTADOS

Para compreender o comportamento de um determinado evento, seja ao longo do tempo, ou em relação a uma variável que tem seu valor alterado, o pesquisador irá se valer de um mecanismo muito eficiente: a observação. Embora a observação seja eficiente, em algumas situações ela não é possível de ser realizada. Para solucionar esse problema, combinamos observação e simulação. A simulação tem sido uma importante ferramenta para a ciência. Os experimentos descritos em seções anteriores foram executados com um número limitado de participantes. Para analisar o comportamento da rede em relação ao número de jogadores conectados, foi necessária a utilização dessa ferramenta, tornando possível a variação de cenários. Cenários que seriam inviáveis em experimentos reais são simulados facilmente.

5.1 Modelagem dos experimentos para o NS2

Antes de se criar os códigos de simulação, que foram executados no NS2, foi necessário modelar os dados observados nos experimentos reais, a fim de se definir os elementos que compõem um determinado cenário. Nessa Seção são apresentados os esquemas gráficos que foram construídos. Esses esquemas possibilitaram a criação de um *Programa-Java* capaz de gerar os *scripts* de simulações que foram utilizados neste trabalho.

Os cenários do experimento 1 são compostos por um *AP*, um servidor do jogo e um número de jogadores que varia para cada fase. No programa de simulação, cada elemento da rede foi representado por um *nó*. O *nó* que representava o *AP* foi configurado como *AP*. No NS2, os elementos de rede são *nós* e os protocolos geradores de eventos agentes. 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 o servidor possui uma conexão com cada um dos jogadores e todos os jogadores se conectam ao servidor. Cada agente de envio do servidor era 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 10.

Para todos os cenários de simulação do experimento 1 e 2, a largura de banda efetiva da rede foi ajustada em 21.7 Mbps (INFO-REVIEWS, 2010) e o tempo de simulação foi igual aos das capturas, de 1200 segundos. A Figura 10 apresenta uma esquema gráfico da arquitetura de construção dos cenários do experimento 1.

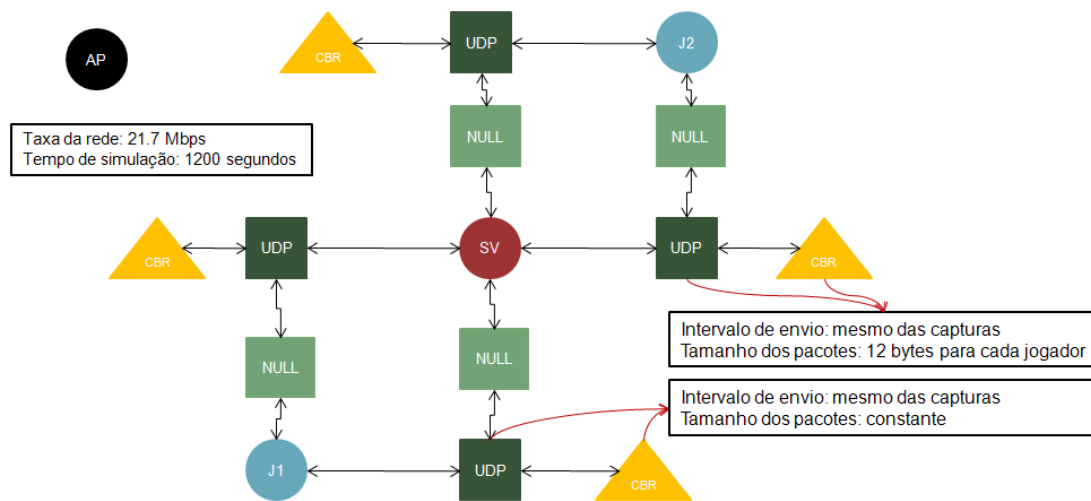


Figura 10: Esquema gráfico de construção do cenário de simulação do experimento 1.

Tanto para o agente de envio quanto para a aplicação, do servidor e do jogo, foram necessários configurar dois parâmetros. O primeiro se refere ao intervalo de envio de pacotes por cada *nó*. Os valores utilizados para essa configuração foram os mesmos valores obtidos nas capturas do experimento 1. O segundo parâmetro diz respeito ao tamanho dos pacotes enviados por cada *nó*. Foi observado, nas capturas, que 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. Relembrando, essas aplicações eram navegação na *WEB*, *streaming* de vídeo e transferência de arquivo, além do jogo do

experimento 1.

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 *default*, 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 11 apresenta como foi construída cada aplicação HTTP nos cenários de simulação.

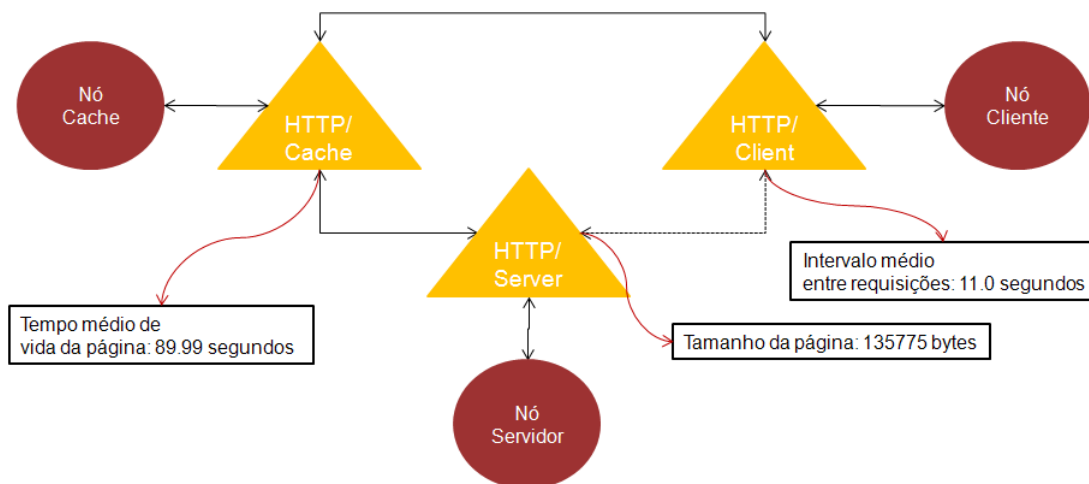


Figura 11: Esquema gráfico de construção da aplicação HTTP.

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 12 apresenta como é construída cada aplicação *streaming* nos cenários de simulação.

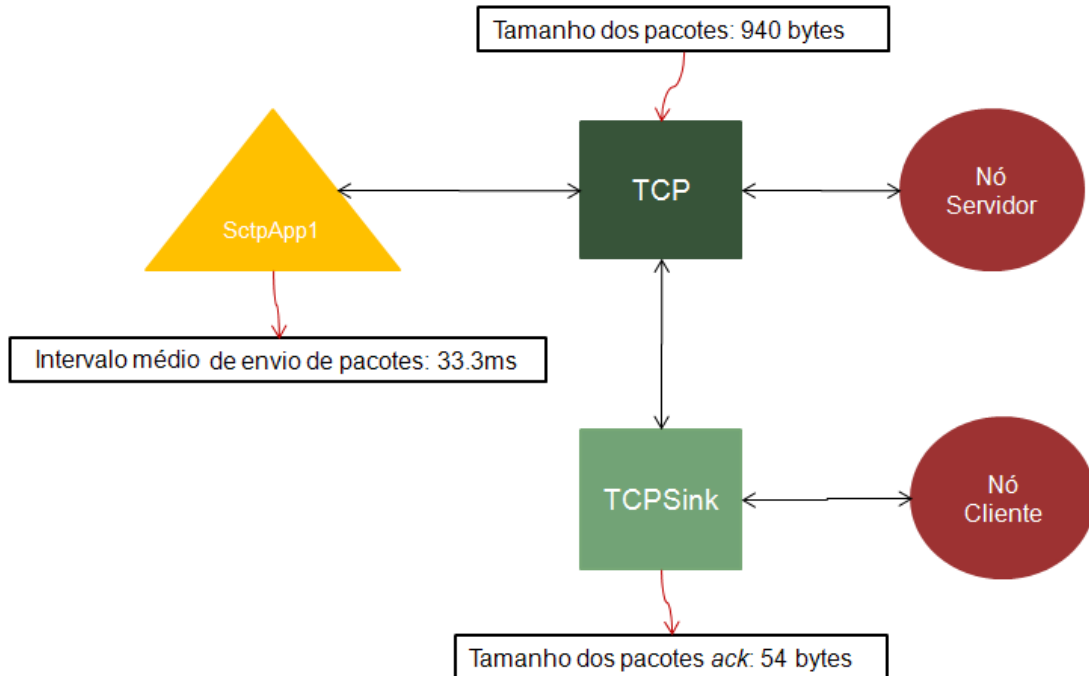


Figura 12: Esquema gráfico de construção da aplicação *streaming*.

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 13 apresenta como é 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*.

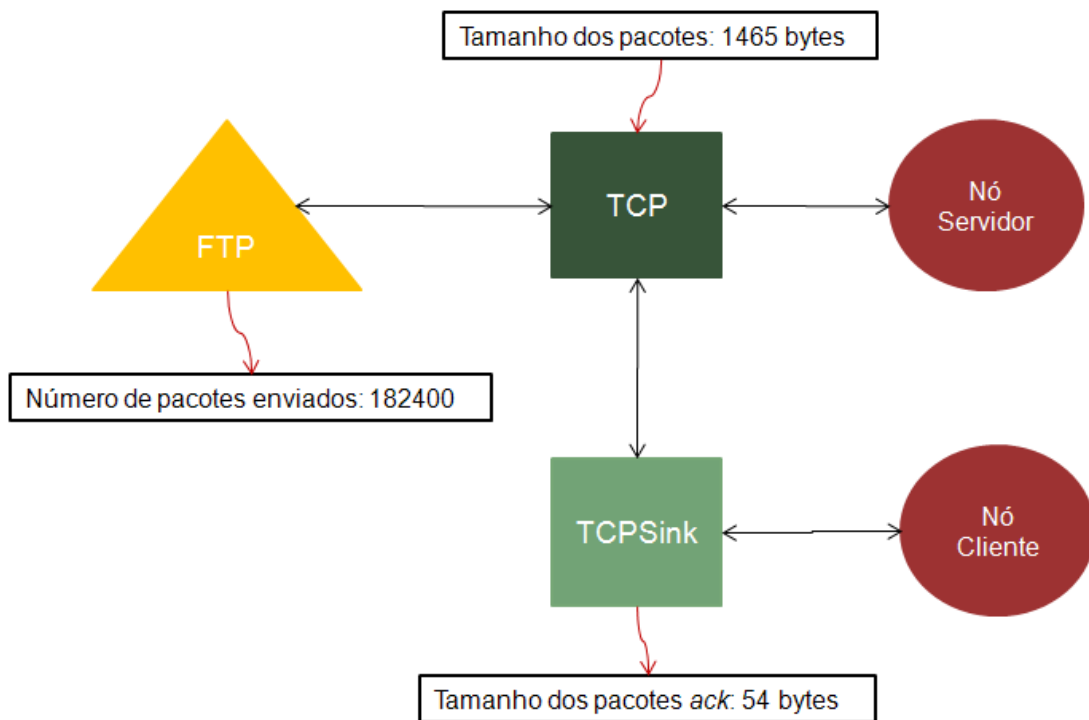


Figura 13: Esquema gráfico de construção da aplicação FTP.

5.2 Gerador de *Scripts* / Analisador

Após o estudo detalhado das capturas e do mapeamento do comportamento de cada *nó* nos experimentos reais, foram geradas diversas tabelas que possibilitaram conhecer e mapear o comportamento da rede, tanto do experimento 1 (somente jogo), quanto do experimento 2 (jogo com outras aplicações). Após as coletas, os valores de intervalo de envio e recebimento de pacotes eram conhecidos, assim como a vazão da rede, o tamanho e a quantidade de pacotes transmitidos.

Foi criado um programa em *Java* para gerar e analisar os dados das simulações. Inicialmente este aplicativo gerou os *scripts* para as fases 1, 2, 3, 4, 5 e 6 do experimento 1. Depois, esses *scripts* foram executados no *NS-2*. Após a execução, os arquivos de saída do *NS-2* foram analisados pelo programa que, por sua vez gera uma série de dados.

O programa *Java-Gerador* constrói *scripts* de simulação baseado no conjunto de dados coletados nos experimentos reais (métricas da rede e do jogo). Para cada fase dos experimentos 1 e 2, foram analisados e catalogados o comportamento do servidor, dos jogadores, e outras aplicações que estavam em execução na rede. Desta forma, uma previsão estatística de comportamentos destes elementos pode ser feita. O programa *Java-Gerador* lia as informações de parâmetros previamente cadastrados e montava os

cenários de simulação (gerar *scripts*). A cada execução do programa eram gerados 10 *scripts*, cada um com suas próprias métricas e configurações que foram selecionadas dos dados coletados nos experimentos reais. Isto significa que cada cenário foi simulado 10 vezes.

5.3 Validação das Simulações - Experimento 1

Após as execuções e análises das saídas do simulador, foi realizado um comparativo entre os dados de saída dos experimentos reais e simulados, com o objetivo de se validar as simulações. Os gráficos das Figuras 14 e 15 mostram, respectivamente os comparativos em relação aos intervalos de envio de pacotes do servidor e dos jogadores no ambiente real e simulado, para a fase 6 do experimento1.

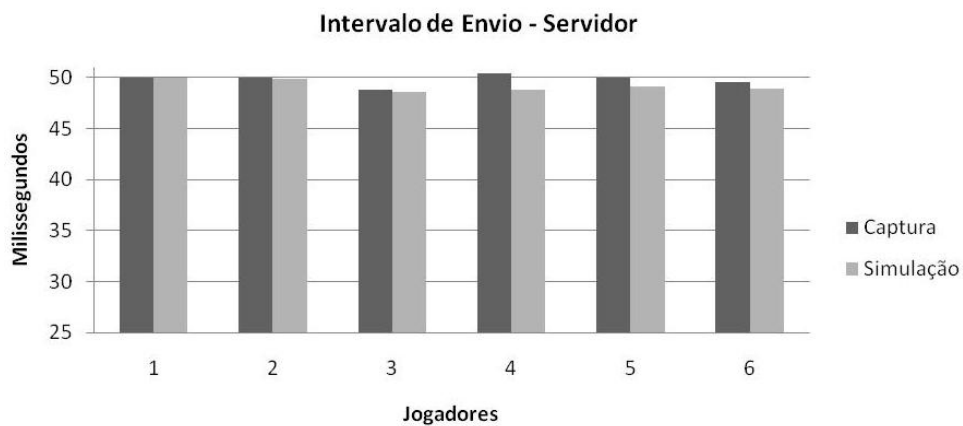


Figura 14: Intervalo de envio de pacotes (SERVIDOR) - Experimentos VS Simulações.

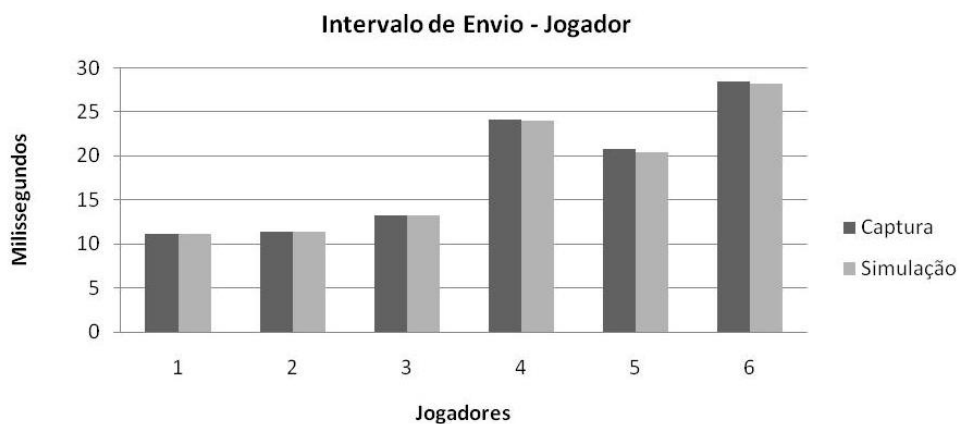


Figura 15: Intervalo de envio de pacotes (JOGADOR) - Experimentos VS Simulações.

Pode-se observar pelos gráficos das Figuras 14 e 15 que, para o parâmetro intervalo de envio, a simulação está validada, pois houve uma pequena diferença entre os dados capturados e os dados extraídos das simulações, dentro dos desvios previstos, conforme é apresentado na tabela 9.

Os gráficos das Figuras 16 e 17, exibem o comparativo do envio de pacotes que trafegaram na rede, nos experimentos reais e nas simulações.

	Capturas	Simulações	Desvio Padrão(Capturas)
Fase 1	50,00	49,96	5,53
Fase 2	50,01	49,86	5,73
Fase 3	48,81	48,61	11,94
Fase 4	50,40	48,78	5,94
Fase 5	50,00	49,06	7,90
Fase 6	49,54	48,85	8,27

Tabela 9: Intervalo de envio (SERVIDOR) - Experimento 1.

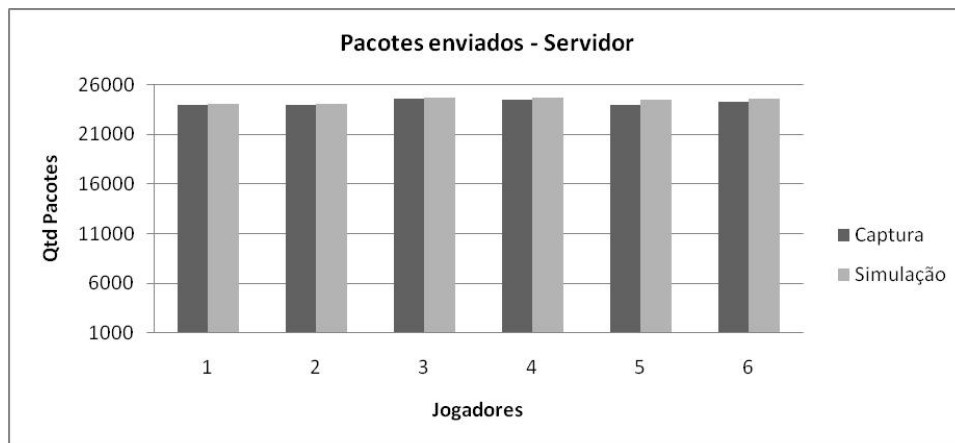


Figura 16: Quantidade de pacotes enviados (SERVIDOR) - Experimentos VS Simulações.

Pode-se observar pelos gráficos das Figuras 16 e 17, que o comportamento das simulações é igual do experimento real. A tabela 10 exibe os desvios padrão aceitáveis. Analisando os desvios padrão dos dados apresentados, conclui-se que as pequenas diferenças observadas estão dentro do esperado.

	Capturas	Simulações	Desvio Padrão(Capturas)
Fase 1	11,18	11,18	2,90
Fase 2	11,34	11,33	3,72
Fase 3	13,30	13,28	7,56
Fase 4	24,07	23,99	6,06
Fase 5	20,72	20,36	7,30
Fase 6	28,47	28,17	11,18

Tabela 10: Intervalo de envio (JOGADOR) - Experimento 1.

Por fim, o consumo de banda da rede, para o experimento 1, para os experimentos reais e simulados, foi praticamente o mesmo, como pode-se observar pelo gráfico da Figura 18.

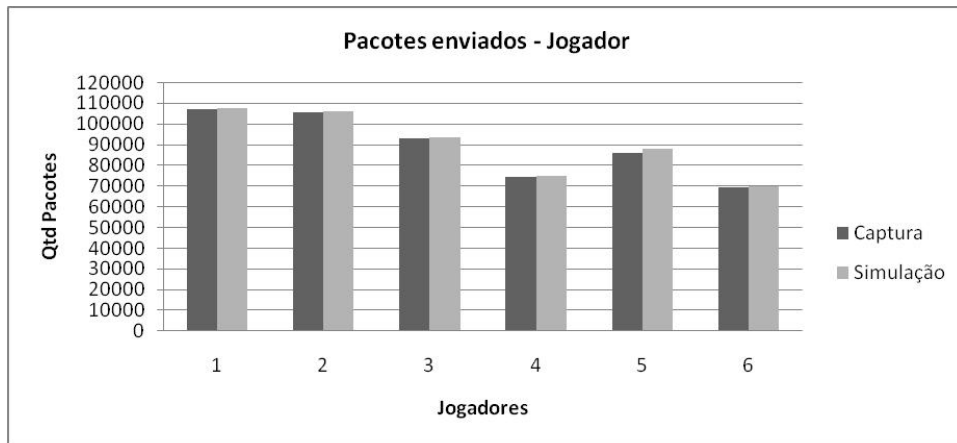


Figura 17: Quantidade de pacotes enviados (JOGADORES) - Experimentos VS Simulações.

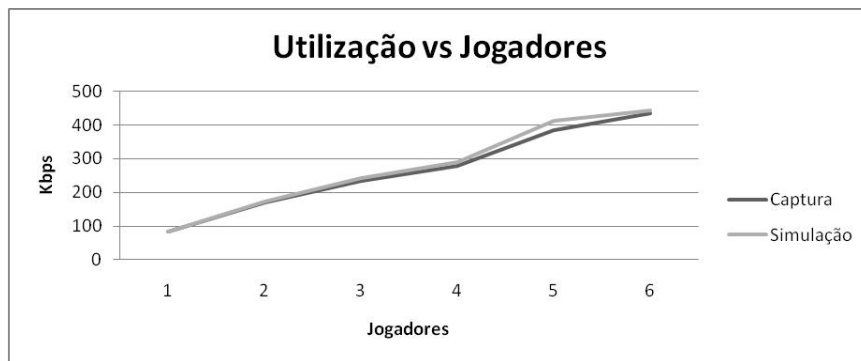


Figura 18: Consumo de banda - Experimento 1 VS Simulações.

5.4 Validação das Simulações - Experimento 2

Para validar o experimento 2, foram construídos três cenários de simulação, enumerados a seguir.

1. 1 Servidor + 6 Jogadores + 1 HTTP
2. 1 Servidor + 6 Jogadores + 1 *Streaming*
3. 1 Servidor + 6 Jogadores + 1 FTP

Os dados de saída das simulações foram avaliados e comparados com os dados obtidos nos experimentos reais. O gráfico da Figura 19 exibe esta validação.

Pode-se observar pelo gráfico da Figura 19, que há uma pequena diferença entre a vazão da rede observada no experimento com relação a vazão observada nas simulações.

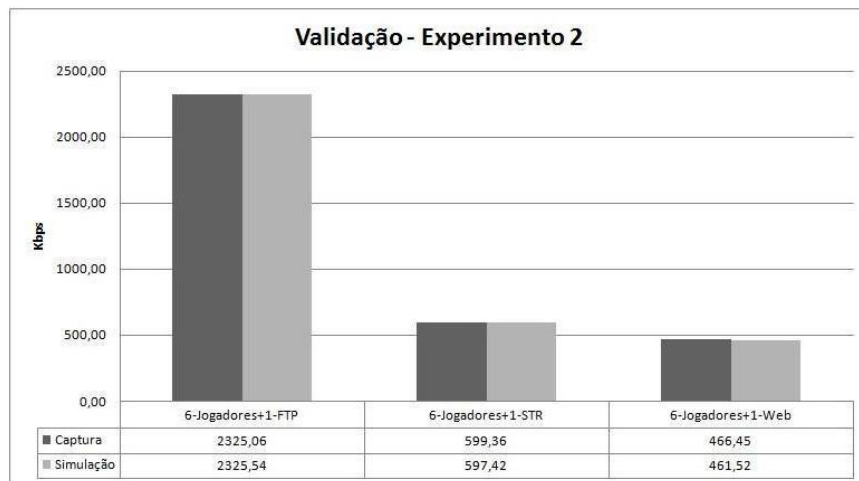


Figura 19: Consumo de banda - Experimento 2 VS Simulações.

Os resultados apresentados estão dentro do previsto e, assim, a simulação foi validada para o experimento 2.

Para esse experimento não foi necessário validar todos os parâmetros do jogo, pois estes já foram validados na Seção 5.3. Os parâmetros das aplicações, como quantidade de pacotes enviados, tempo de requisição e largura de banda consumida foram semelhantes ao observados nos experimentos reais.

Ao termino da validação do experimento 1 e 2, temos a possibilidade de criar novos cenários a fim de identificar pontos críticos que possam responder às questões apresentadas no início deste trabalho.

5.5 Simulações de novos cenários - Experimento 1

Nos experimentos reais, foram analisados os parâmetros em relação à execução de um jogo *FPS* com seis jogadores, conectados a uma rede *Wi-Fi*. Foi observado que este número de jogadores não causa grandes impactos na rede, e que a jogabilidade manteve-se satisfatória em relação ao intervalo de recebimento de pacotes, que deve ser inferior a 150 ms (FARBER, 2002).

A simulação de novos cenários permite inserir jogadores ao ambiente, até que esse limite de 150 ms seja alcançado. Desta forma, a primeira questão levantada neste trabalho poderá ser respondida, ou seja, identificar a quantidade de jogadores que podem ser inseridos na rede sem prejuízo aos jogadores que já estão conectados à rede. Foram realizadas simulações de novos cenários com 1 jogador e, a cada interação, mais 1 jogador

foi acrescentado ao ambiente. O gráfico da Figura 20 mostra o comportamento da rede para as execuções das simulações do experimento 1, desde 1 jogador até 20 jogadores, na rede. Conforme é apresentado no gráfico da Figura 20, pode-se notar que à medida que se expande o cenário, inserindo um novo jogador, o consumo de banda aumenta, como era de se esperar. Pode-se observar que, ao se atingir 15 jogadores na rede, o consumo de banda se estabiliza. Isso ocorre porque a partir desse cenário a rede está degradada e, há uma alta taxa de perda de pacotes.

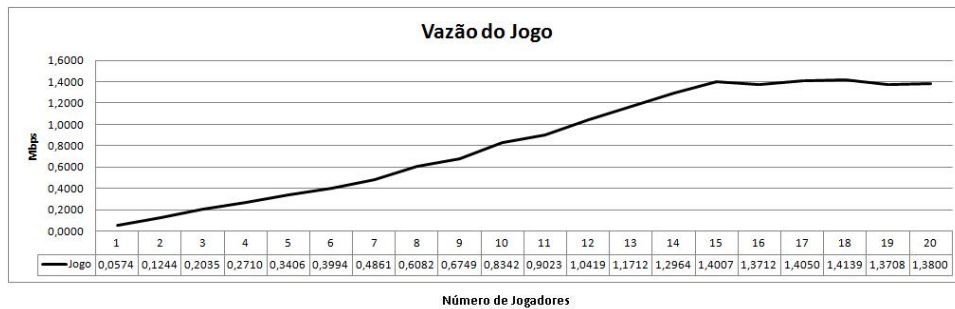


Figura 20: Vazão total do jogo de 1 jogador até 20 jogadores.

Como o parâmetro de qualidade para o jogo é o intervalo de envio e recebimento, é necessário se verificar o que ocorre com estes parâmetros quando há 15 jogadores na rede. Esses parâmetros são apresentados nos gráficos das Figuras 21 e 22.

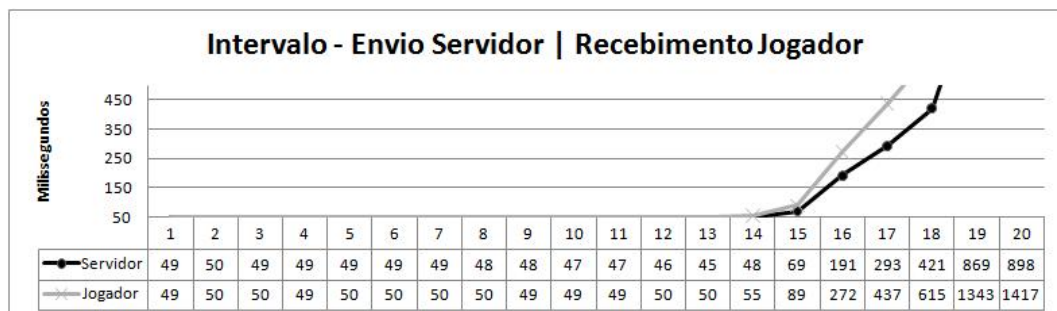


Figura 21: Intervalo de envio de pacotes do servidor e recebimento no jogador.

O gráfico da Figura 21 mostra que, a partir de 16 jogadores, o limite definido como característica principal para uma boa jogabilidade é ultrapassado. Com 16 jogadores na rede, tem-se um intervalo de envio do servidor em torno de 191 ms e recebimento no jogador de 272 ms. Esses valores dificultam a jogabilidade, porque o jogador tem um efeito desagradável denominado *lag* (atraso). Já o intervalo do jogador se manteve dentro dos parâmetros aceitáveis.

Assim percebeu-se que a jogabilidade é prejudicada a partir de 16 jogadores na

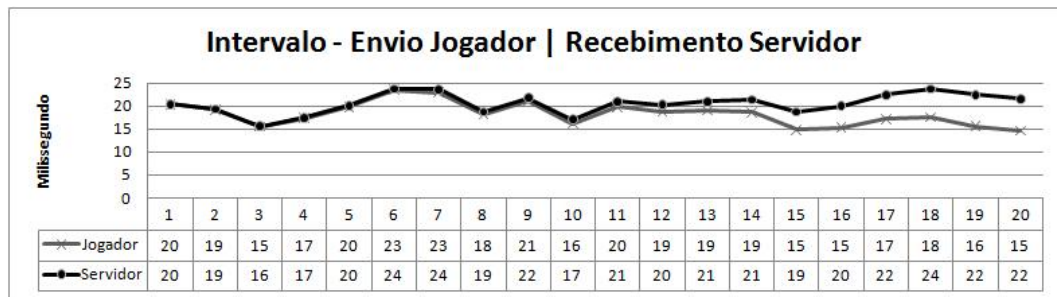


Figura 22: Intervalo de envio de pacotes do jogador e recebimento no servidor.

rede. Esta simulação gerou um questionamento que precisava ser respondido antes de se prosseguir com as simulações. Foi a rede que não suportou mais de 16 jogadores conectados ou foi o fato de haver apenas um servidor que gerou o problema, ou seja, o servidor foi o gargalo da rede? Para resolver esta questão, foram simulados cenários com 32 jogadores e um servidor e 32 jogadores e dois servidores, sendo que, para este último cenário, havia 16 jogadores conectados em cada servidor. Foi observado que, tanto para o cenário com um servidor como para o cenário com dois servidores, não houve grande alteração quanto à capacidade da rede em transferir dados. Com este teste, conseguiu-se um acréscimo de apenas 0.15 Mbps. O que mostra que a divisão da população de jogadores em dois servidores não foi capaz de aumentar o tráfego da aplicação do jogo na rede, confirmando o número máximo de 15 jogadores com qualidade. O gráfico da Figura 23 mostra a perda de pacotes do servidor e dos jogadores, à medida que mais jogadores são inseridos na rede. Pode-se notar que, a partir de 15 jogadores, há um aumento considerável na perda de pacotes.

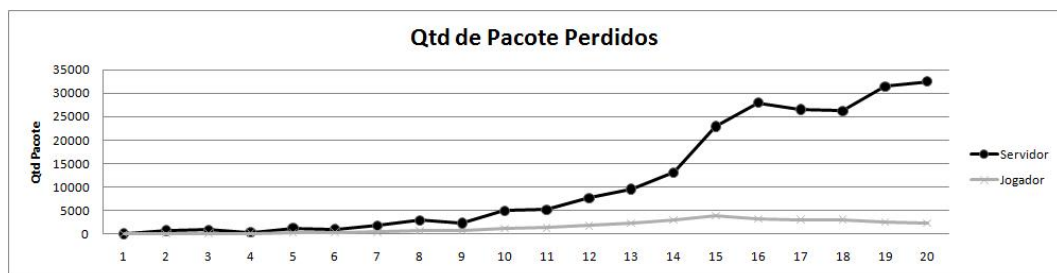


Figura 23: Quantidade de pacotes perdidos.

5.6 Simulações de novos cenários - Experimento 2

No experimento 2, para se verificar o comportamento do jogo e das aplicações, foram criadas quatro fases nas simulações. Essas fases foram construídas da seguinte

forma:

- Fase 1 - Somente um tipo aplicação.

Nessa fase, havia somente a aplicação rodando na rede. Neste momento, as aplicações não estavam concorrendo com o jogo e nem com outros tipos aplicações. Sendo assim, tínhamos simulações com apenas a aplicação HTTP rodando, e a cada simulação o número de *nós* executando a aplicação era acrescido até o limite definido de 20 *nós*. Esta estrutura de simulação foi utilizada também pelas aplicações *streaming* e FTP. O objetivo aqui era verificar qual é o impacto que a aplicação causa na rede e nela mesma.

- Fase 2 - Fixar 6 jogadores na rede e colocar um tipo de aplicação, e variar o número de *nós* da aplicação de 1 até 20.

Nessa fase, havia 6 jogadores na rede e um tipo de aplicação rodando, por exemplo: foram inserido 6 jogadores na rede e um *nó* com a aplicação HTTP, depois foram inseridos mais *nós* com a aplicação HTTP até 20 *nós* com a aplicação e os 6 jogadores. Esta estrutura de simulação foi utilizada também pelas aplicações *streaming* e FTP. O objetivo aqui era verificar se a aplicação impacta na qualidade do jogo e em que ponto isso ocorria.

- Fase 3 - Fixar as aplicações e aumentar o número de jogadores na rede, até 15 jogadores. Nessa fase, havia um número pré-estabelecido de aplicações sendo executadas na rede. Este número se baseou no limite observado na fase 2, que determinou que 4 aplicações FTP, 6 HTTP e 6 *streaming* não afetaram a rede. O objetivo aqui era verificar se o jogo impacta na qualidade das aplicações e em que ponto isso ocorria.
- Fase 4 - Fixar o jogo em 6 jogadores e inserir *nós* (misturar as aplicações) com as aplicações FTP, HTTP e *streaming*.

Como foram observados nos experimentos reais e nas simulações, 6 jogadores na rede não tem um grande impacto, a jogabilidade se mantém dentro dos parâmetros aceitáveis. Nessa fase, foram criados diversos cenários com 6 jogadores conectados que estavam concorrendo com as aplicações. Desta forma, foi possível observar quando as combinações de aplicações interferem na jogabilidade. Foram criados cenários com 6 jogadores e três grupos de *nós*, cada um executando uma quantidade específica de aplicações. Como exemplo: Tínhamos um cenário com 6 jogadores e um grupo com três *nós* executando as aplicações (*1 HTTP e 1 streaming e 1 FTP*), depois foi simulado um cenário com 6 jogadores e 6 *nós* (*2 HTTP e 2 streaming e*

2 FTP) até o último cenário com 6 jogadores e 12 nós rodando aplicação (4 HTTP e 4 streaming e 4 FTP).

5.6.1 Fase 1 - Somente Aplicação

Para a fase 1, foram construídos cenários que estavam executando apenas cada aplicação. As aplicações simuladas foram as mesmas que foram executadas nos experimentos reais. A cada interação das simulações o número de nós na rede era acrescido de um. O limite definido para o número de nós na rede foi 20. O gráfico da Figura 24 apresenta as simulações para a aplicação HTTP.

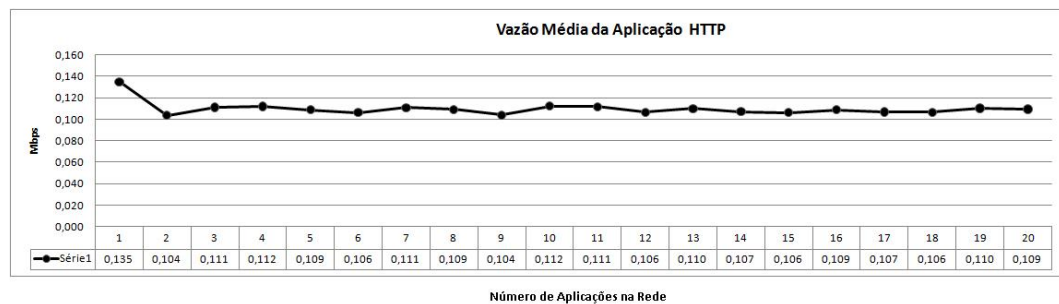


Figura 24: Vazão média da aplicação até 20 HTTP.

Pode-se observar, pelo gráfico da Figura 24, que a vazão média da aplicação HTTP se manteve estável para todas as simulações. Como pode ser observado, a vazão média da aplicação foi de 104,28 Kbps, com desvio padrão de 6,43 Kbps. O aumento no número de usuários não acarretou congestionamento na rede.

Para as simulações da aplicação FTP, os dados referentes à vazão média da rede e à perda de pacotes são apresentados nos gráficos das Figuras 25 e 26, respectivamente. Por eles, observa-se que, a partir de três usuários executando a aplicações FTP, há degradação na rede. À medida que colocamos mais nós, essa degradação fica mais evidente. O gráfico da Figura 26, mostra que, à medida em que novos nós foram incorporados à rede, o percentual de perda de pacotes aumentam. Para aplicações FTP, esse é o parâmetro que deve ser analisado, pois FTP é extremamente sensível a perdas.

Para as simulações de *streaming*, a cada interação, um nó com a aplicação era inserido, até a última simulação que continha 20. A vazão da aplicação se manteve estável, em torno de 252,31 Kbps, com desvio padrão de 24,72 Kbps. *Streaming* é sensível a *jitter*, não será apresentado um gráfico para esta informação, uma vez que para todos os cenários esse parâmetro se manteve abaixo de 1 ms.

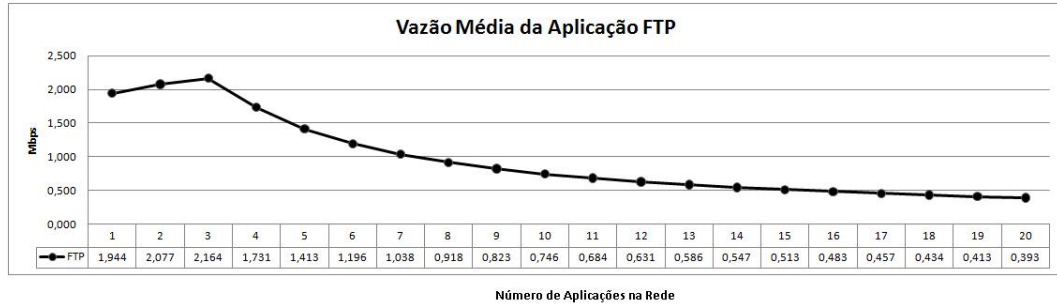


Figura 25: Vazão média da aplicação até 20 FTP.

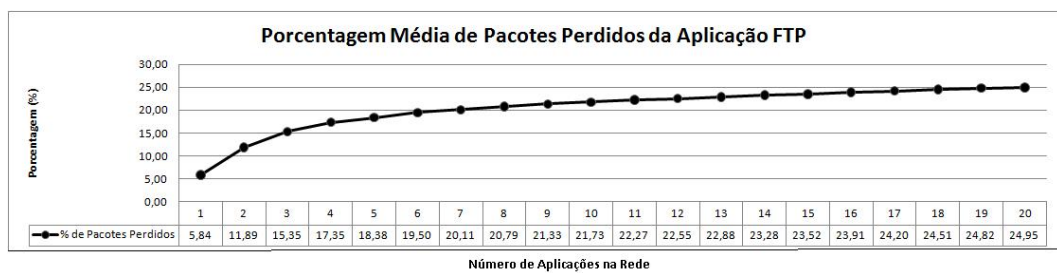


Figura 26: Percentual de perda de pacotes da aplicação até 20 FTP.

5.6.2 Fase 2 - Simulações com 6 Jogadores e Aplicações

Nas seções anteriores, pode-se verificar o impacto que uma determinada aplicação pode causar na rede ou em outras aplicações. Nessa seção será avaliado o quanto a aplicação impacta no jogo. Sendo assim, foram construídos cenários com 6 jogadores (número fixo) e diversas aplicações sendo executadas no ambiente de simulação.

O gráfico da Figura 27 mostra os resultados das simulações com 6 jogadores e a aplicação HTTP. Foram construídos, inicialmente cenários com seis *nós* jogadores e um *nó* com a aplicação HTTP. Depois, o número de *nós* executando a aplicação foi acrescido de um, até o último cenário que tinha 6 jogadores e 20 *nós* executando HTTP.

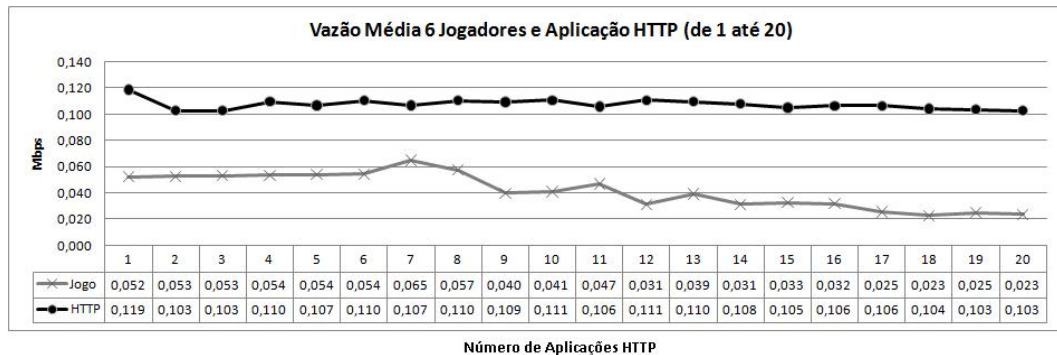


Figura 27: Vazão média de 6 jogadores e da aplicação até 20 HTTP.

Conforme observou-se pelo gráfico da Figura 27, a aplicação HTTP se manteve estável ao longo das simulações. Esse comportamento também foi observado quando analisamos o comportamento da aplicação sendo executada sem concorrência na rede. Aqui se pode verificar que 6 jogadores não interferem na qualidade das aplicações HTTP, pois sua vazão permaneceu dentro dos parâmetros aceitáveis. Com relação à qualidade do jogo, mesmo com o aumento da qualidade de aplicações HTTP na rede, o intervalo de envio e recebimento de pacotes se manteve dentro dos parâmetros aceitáveis. Assim conclui-se que este número de aplicações HTTP na rede não foi suficiente para prejudicar a jogabilidade. O intervalo de envio do servidor ficou abaixo dos 53 ms e do jogadores em 72 ms.

Simulações semelhantes foram realizadas para se verificar o impacto que a aplicação *streaming* causa no jogo. O gráfico da Figura 28 mostra a vazão média da rede, para os 6 jogadores e para as 20 aplicações *streaming*. Ele mostra o comportamento da vazão das aplicações *streaming* e do jogo, à medida que novos *nós* com a aplicação foram inseridos na rede. Pode-se observar que, a partir de 10 aplicações *streaming* na rede, há uma perda

gradual na vazão do jogo e este ponto indica um limite. Para verificar em que ponto se encontra o limite que cause o prejuízo a jogabilidade, precisamos analisar as informações referentes aos intervalos de envio e recebimento do servidor e dos jogadores. Os gráficos das Figuras 29 e 30, exibem os intervalos de envios e recebimentos de pacotes do servidor e dos jogadores para estes cenários. Pode-se observar, no gráfico da Figura 30, que o intervalo de envio do servidor fica abaixo dos parâmetros pré-estabelecidos (150 ms). No gráfico da Figura 29, percebe-se claramente que, a partir de 15 aplicações *streaming* na rede, esse parâmetro não é mais respeitado. Isso nos mostra que, se houver um ambiente com 6 jogadores e 15 aplicações *streaming* sendo executados, certamente essas aplicações irão impactar na qualidade do jogo, no que se refere ao tempo de resposta e prejudicará a experiência de jogo para o jogador.

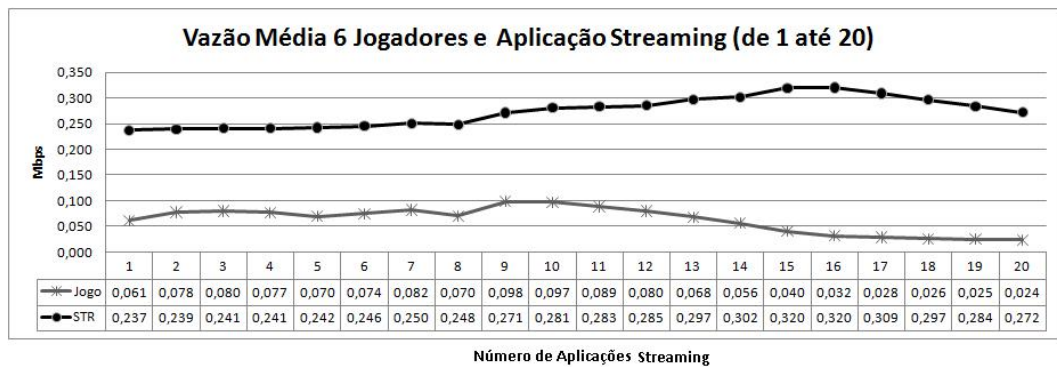


Figura 28: Vazão média de 6 jogadores e da aplicação até 20 *streaming*.

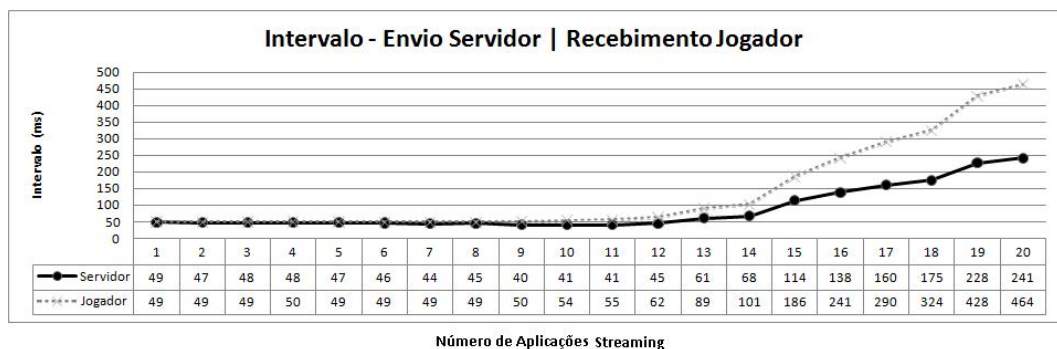


Figura 29: Intervalo de envio de pacotes do servidor e recebimento no jogador.

Para avaliar o real impacto que a aplicação causou no jogo, o gráfico da Figura 31 mostra o comportamento do jogo sendo executado sozinho e o comportamento do jogo quando este estava concorrendo com a aplicação *streaming*. Antes de se explicar os dados do gráfico propriamente dito, vale ressaltar mais uma vez o comportamento dos jogadores no ambiente de simulação. Nos experimentos reais, havia de um até seis

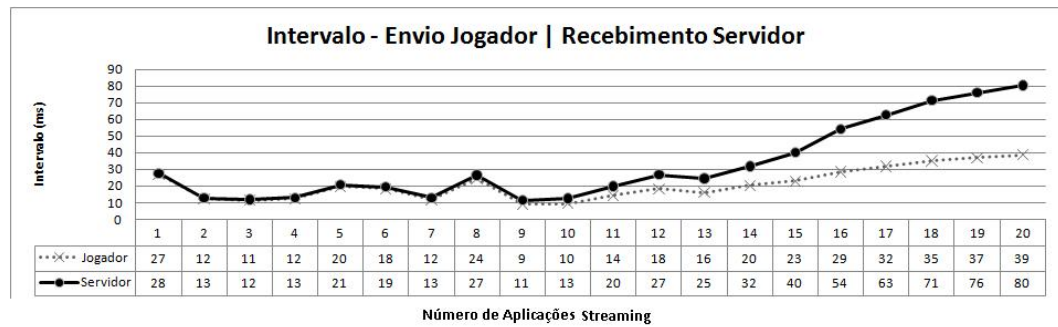


Figura 30: Intervalo de envio de pacotes do jogador e recebimento no servidor.

jogadores na rede e os comportamentos desses jogadores foram catalogados e utilizados como parâmetros de entrada para a nossa simulação. A partir desse ponto, essa amostra de comportamento serviu para gerar o comportamento de todos os jogadores na rede. Assim, o comportamento desses seis jogadores foi inserido em uma função, no programa *Java-Gerador* e, toda vez que um cenário era gerado, o gerador selecionava aleatoriamente o comportamento que cada jogador teria (intervalo de envio e recebimento de pacote) e atribua ao *nó* do jogador. Sendo assim, podemos ter, por exemplo, um cenário com seis jogadores que tiveram uma vazão média de 0.68 Mbps, e em outro cenário com o mesmo número de jogadores temos a vazão média em torno de 0,60 Mbps. Isto pode ser explicado da seguinte forma. No nosso vetor de comportamento, temos o comportamento de *notebooks* com maior e menor desempenho, como foi observado nos experimentos reais e, portanto, em determinado momento, a função aleatória seleciona mais características boas do que ruins ou vice-versa.

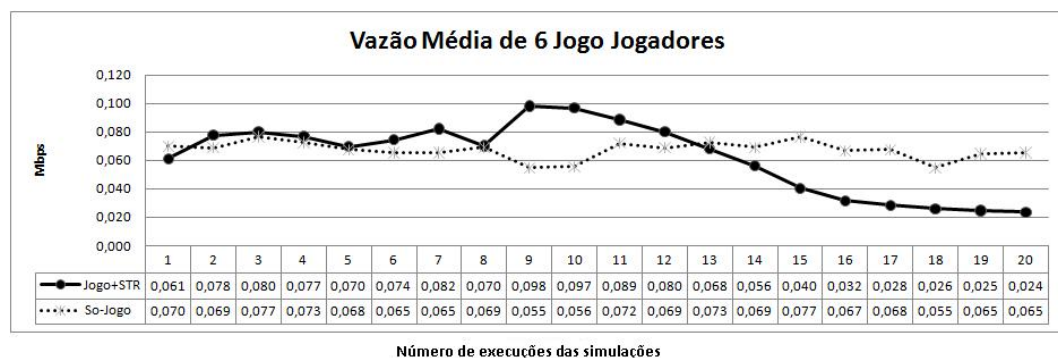


Figura 31: Vazão média de 6 jogadores sozinhos na rede e 6 jogadores e *streaming*.

Concluindo, pode-se observar no gráfico da Figura 31 que, como já era previsto, com 15 aplicações *streaming* rodando na rede, a vazão da aplicação do jogo cai consideravelmente.

Por fim, analisou-se o impacto que a aplicação FTP causa no jogo. Essas simulações, seguiram a mesma metodologia usada para as aplicações HTTP e *streaming*.

O gráfico da Figura 32 apresenta a vazão média da rede ao longo de cada interação das simulações com FTP. Pode-se observar pelo gráfico da Figura 32, que há um grande impacto da aplicação FTP, tanto nos outros *nós* que estão executando a aplicação quanto no jogo. Pode-se notar que, a partir de 8 aplicações FTP conectadas na rede, há uma perda por parte do jogo na sua capacidade de transmitir dados, pois sua vazão média começa a diminuir. Verifica-se, que os intervalos de envio e recebimento de pacote começam a subir, quando há 8 aplicações FTP sendo executadas na rede. O parâmetro de qualidade para o jogo deixa de ser respeitado, indicando que, a partir desse ponto, a jogabilidade é afetada.

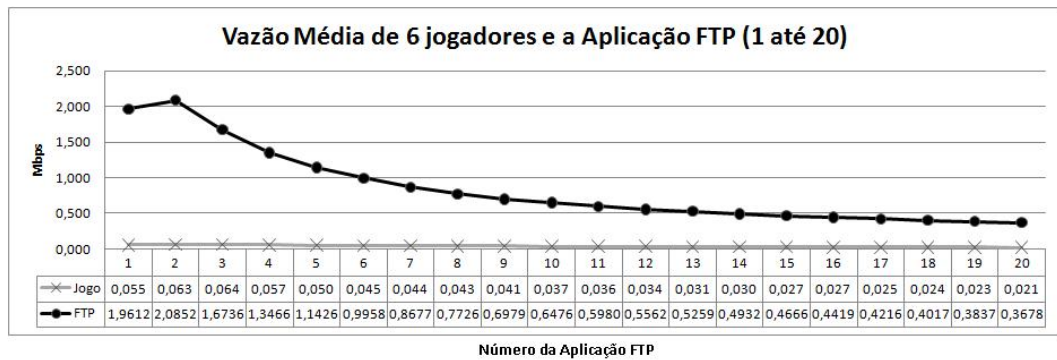


Figura 32: Vazão média de 6 jogadores e da aplicação até 20 FTP.

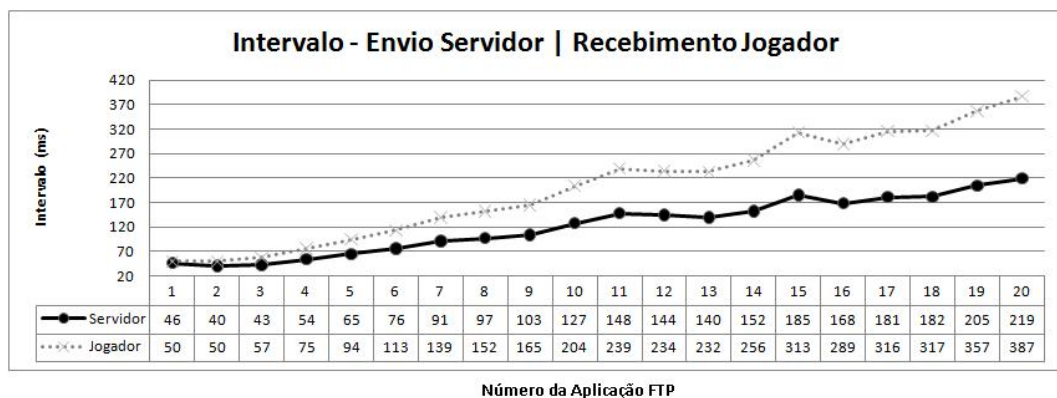


Figura 33: Intervalo de envio de pacotes do servidor e recebimento no Jogador.

Para se verificar o verdadeiro impacto que a aplicação causa no jogo, o gráfico da Figura 34 exhibe a vazão média de 6 jogadores sozinhos na rede e de 6 jogadores mais a aplicação FTP.

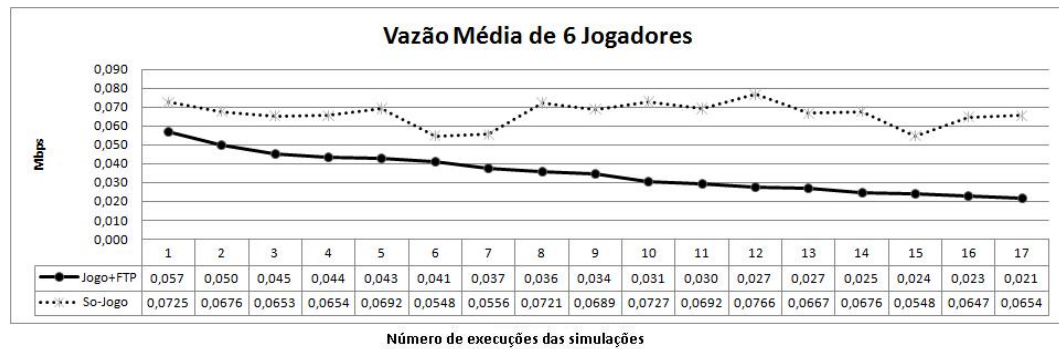


Figura 34: Vazão média de 6 jogadores sozinhos na rede e 6 jogadores e FTP.

Confirmando análise dos gráficos anteriores, observa-se que, a partir de 8 aplicações na rede, o jogo começa a ser degradado de uma forma que a sua jogabilidade é comprometida.

5.6.3 Fase 3 - Simulações Aplicação Fixa e Jogadores até 15

Para os cenários apresentados nesta seção foram avaliados os impactos que o jogo pode causar nas aplicações. Para realizar tais simulações, o número de nós que executava uma aplicação se manteve constante e o número de jogadores variou, a cada iteração. O número fixo de aplicações na rede foi retirado das análises anteriores, que mostraram limites em que o jogo e as aplicações não estavam impactando a rede. Sendo assim, para a aplicação HTTP, foram construídos cenários com 6 aplicações e o número de jogadores variando de 1 até 15 jogadores.

O número máximo de jogadores foi definido como 15, limite máximo observado nas simulações anteriores, onde a rede suporta a carga sem degradar a qualidade do próprio jogo. Os gráficos das Figuras 35 e 36 mostram o comportamento analisado nas simulações para o HTTP e para o jogo. Pode-se observar que a vazão média da aplicação e do jogo se mantiveram estáveis, dentro dos parâmetros de qualidade. Desta forma, foi observado que as 6 aplicações HTTP não foram afetadas pelo número máximo de jogadores conectados à rede. Analisando o gráfico da Figura 36, percebe-se que o jogo também não foi afetado e que os parâmetros de intervalo de envio estão dentro do aceitável. Não será mostrado o gráfico do intervalo de envio do jogador, pois este ficou abaixo dos 50 ms.

Dando continuidade às simulações da fase 3, os gráficos das Figuras 37 e 38 apresentam o comportamento das simulações para 6 aplicações *streaming* e jogadores conectados à rede entre 1 e 15. Conforme observado no gráfico da Figura 37, a vazão média da aplicação *streaming* e do jogo se mantiveram constantes. Existem algumas variações, mas elas

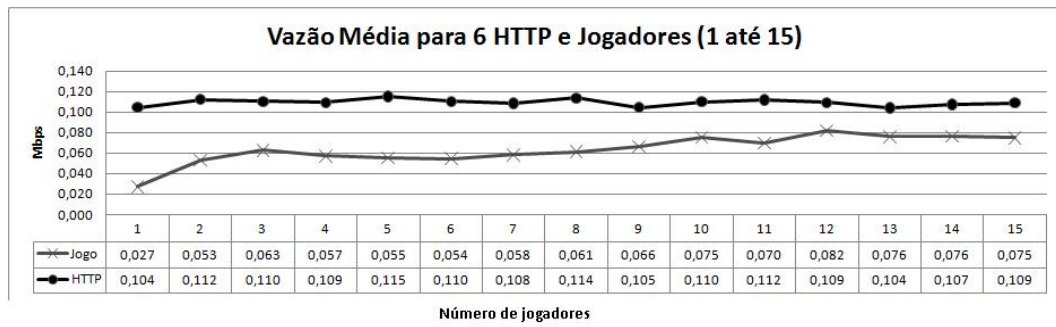


Figura 35: Vazão média de 6 aplicações HTTP e jogadores até 15.

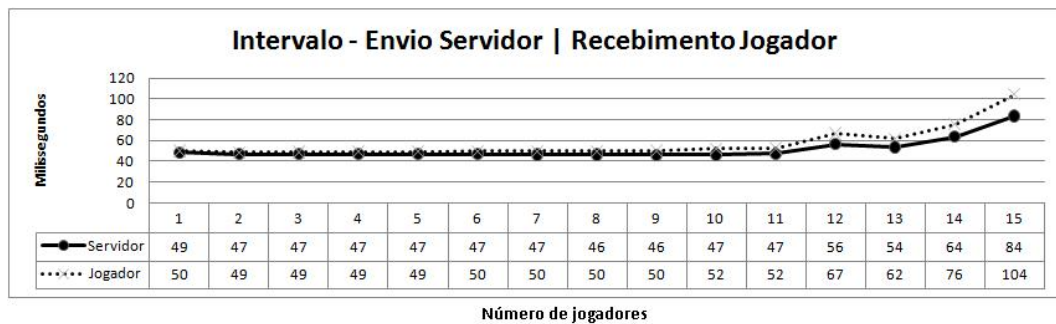


Figura 36: Intervalo de envio do servidor e recebimento no jogador - 6 HTTP e jogadores até 15.

não afetaram a qualidade da aplicação. Seguindo recomendações da literatura, o parâmetro de qualidade para a aplicação *streaming* avaliado neste trabalho é o *jitter*. Em todas as simulações realizadas o *jitter* se manteve abaixo de 1 ms. O gráfico da Figura 38 mostra o intervalo de envio do servidor e de recebimento de pacotes no jogador. Comparando este gráfico com o gráfico da Figura 21 pode-se perceber que o parâmetro de qualidade para o intervalo de envio se elevou com 12 jogadores, o que nos indica que as 6 aplicações de *streaming* tiveram impacto no jogo.

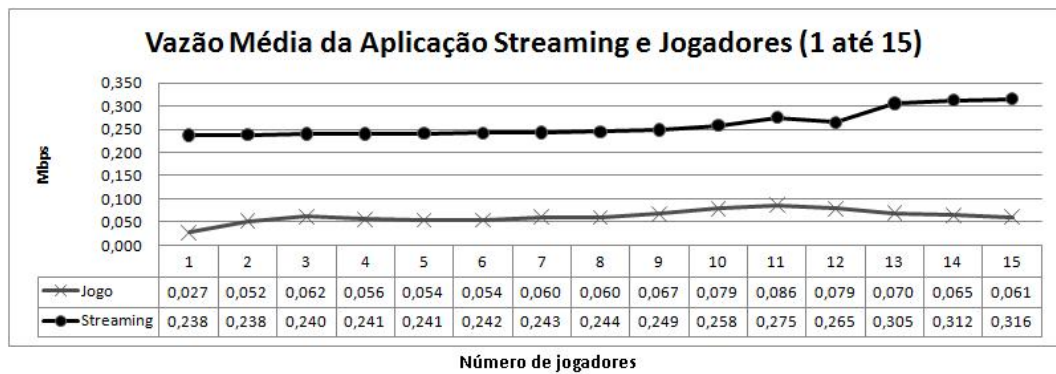


Figura 37: Vazão média de 6 aplicações *streaming* e jogadores até 15.

Por fim, na fase 3, foi avaliado o possível impacto que o jogo pode causar na

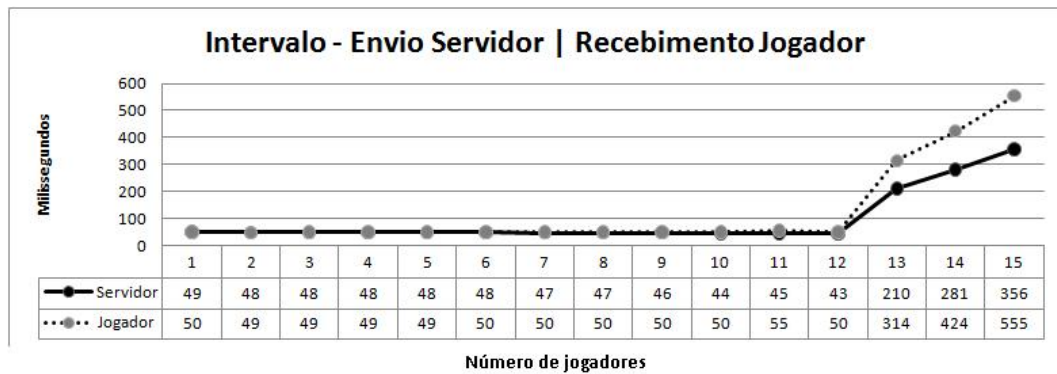


Figura 38: Intervalo de envio do servidor e recebimento no jogador - 6 *streaming* e jogadores até 15.

aplicação FTP. Esses cenários de simulação foram construídos com 4 aplicações FTP e o número de jogadores variando de 1 até 15. A quantidade de aplicações na rede foi determinada porque este número de aplicações não causar grandes impactos no jogo e na rede, como já foi demonstrado. Conforme observado no gráfico da Figura 39 o tráfego da aplicação FTP estava tendo impacto a cada novo jogador que ingressava na rede. A vazão média de 4 aplicações FTP (somente aplicação) sendo executada na rede foi de 1.73 Mbps. Com o jogo, 4 FTP's não conseguiram transferir a essa taxa e, a cada iteração, o volume trafegado pela aplicação diminuiu. Aqui se vê que o jogo impactou diretamente na aplicação. Analisando o comportamento do jogo também foi observado que 4 aplicações FTP causaram um impacto no jogo. Inicialmente, para manter QoE , verificou-se que o limite de jogadores na rede era de 15 jogadores, mas com a execução de 4 aplicações FTP este limite foi reduzido para 8 jogadores, conforme é mostrado no gráfico da Figura 40.

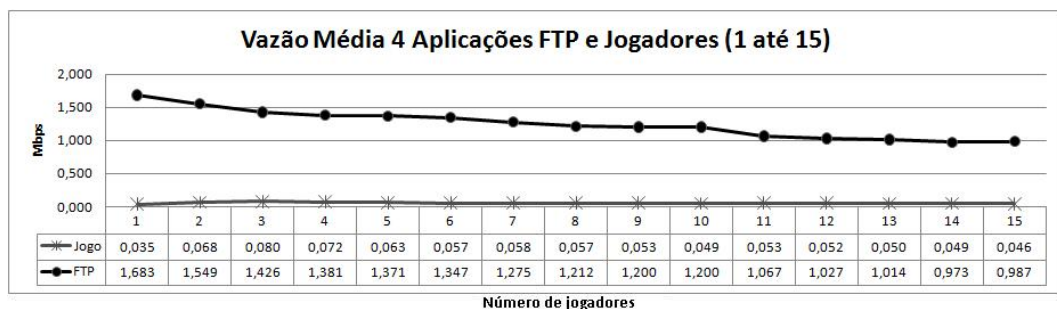


Figura 39: Vazão média de 4 aplicações FTP e jogadores até 15.

Para se validar este impacto causado pelo jogo na aplicação FTP, foi avaliado o parâmetro de QoS para o FTP, que é o percentual de perda de pacotes na rede. Conforme é apresentado no gráfico da Figura 41, pode-se observar que o percentual de perda é maior quando o FTP está concorrendo com o jogo. Pode-se observar, pelo gráfico da Figura 41

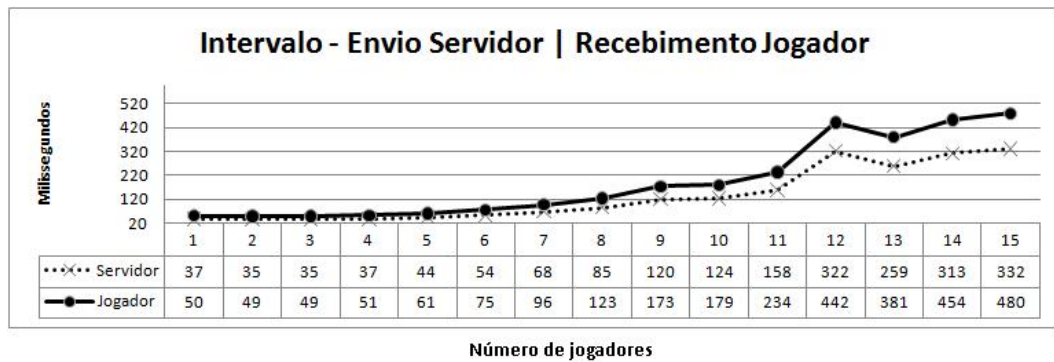


Figura 40: Intervalo de envio do servidor e recebimento no jogador - 4 FTP e jogadores até 15.

que, à medida que mais jogadores entram na rede, o percentual de perda de pacotes da aplicação aumenta, o que indica o impacto que o jogo tem sobre a aplicação FTP.

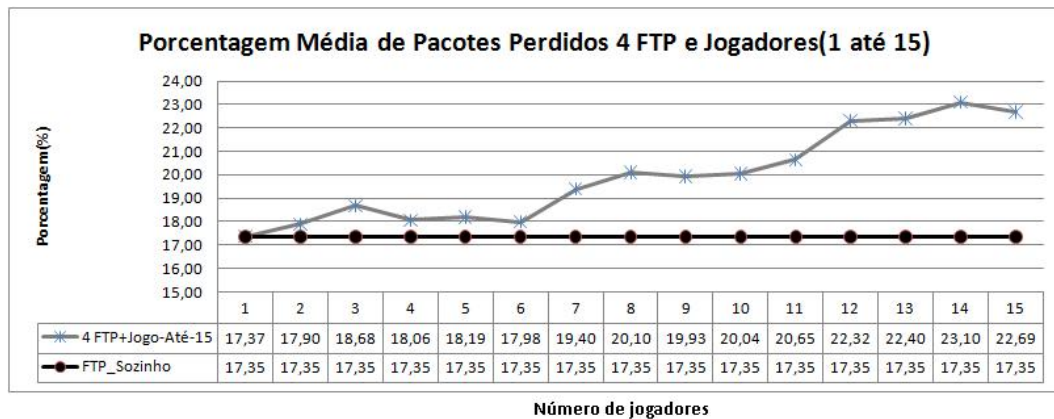


Figura 41: Percentual de perda de pacotes para a aplicação FTP.

5.6.4 Fase 4 - Simulações de Cenários Mistos

Nesta fase foi analisado o comportamento do jogo e de outras aplicações simulados simultaneamente. As quantidades de elementos na rede foram extraídas das observações realizadas até o momento. A tabela 11 mostra como foram montados estes cenários de simulações.

	Jogo	HTTP	streaming	FTP
Cenário-6111	6	1	1	1
Cenário-6222	6	2	2	2
Cenário-6333	6	3	3	3
Cenário-6444	6	4	4	4

Tabela 11: Cenários da fase 4, jogo e outras aplicações.

Observa-se na tabela 11, que foram criados diversos cenários para se avaliar o comportamento do jogo e das aplicações. Para essas simulações, todas as aplicações estavam sendo executada junto com o jogo. Os cenários foram compostos por diversos *nós* executando o jogo e as aplicações. Foi criada uma legenda para estes cenários. A legenda Cenário-JHSF pode ser lida da seguinte forma: Cenário-JHSF onde cada letra representa uma quantidade de *nós* na rede e um tipo de aplicação: Onde J é o número de jogadores, H é o número de aplicações HTTP, S é o número de aplicações *streaming* e F é o número de aplicações FTP. Como exemplo, o cenário *Cenário-6111* foi composto da seguinte forma:

- J=6 -> temos 6 jogadores
- H=1 -> temos 1 aplicação HTTP
- S=1 -> temos 1 aplicação *Streaming*
- F=1 -> temos 1 aplicação FTP

O *Cenário-6444* foi composto da seguinte maneira:

- J=6 -> temos 6 jogadores
- H=4 -> temos 4 aplicação HTTP
- S=4 -> temos 4 aplicação *streaming*
- F=4 -> temos 4 aplicação FTP

O objetivo aqui foi o de identificar combinações que nos mostrassem impactos causados no jogo que estivessem relacionados com algum tipo de combinação. Para essas simulações, são apresentados os gráficos que representam os parâmetros de *QoS* para cada aplicação. Desta forma, pode-se ter uma visão mais objetiva dos dados apresentados para cada cenário executado, em relação aos parâmetros que se deseja analisar.

O gráfico da Figura 42 apresenta a vazão média para o aplicação HTTP. Pode-se notar que para todas as combinações construídas a vazão média da aplicação HTTP se manteve dentro dos parâmetros pré-estabelecidos, o que indica que, para esta aplicação, as combinações geradas não causaram impacto. Para a aplicação *streaming* o parâmetro analisado deve ser o *jitter*, este parâmetro não sofreu impacto e para os cenários realizados

o *jitter* se manteve abaixo de 1 ms. Já a aplicação FTP, é a que, até o momento, demonstrou-se mais instável, seja em relação ao número de *nós* com a aplicação FTP na rede, seja com relação à aplicação concorrendo com o jogo. O gráfico da Figura 43 nos mostra como se comportou esta aplicação para os cenários construídos.

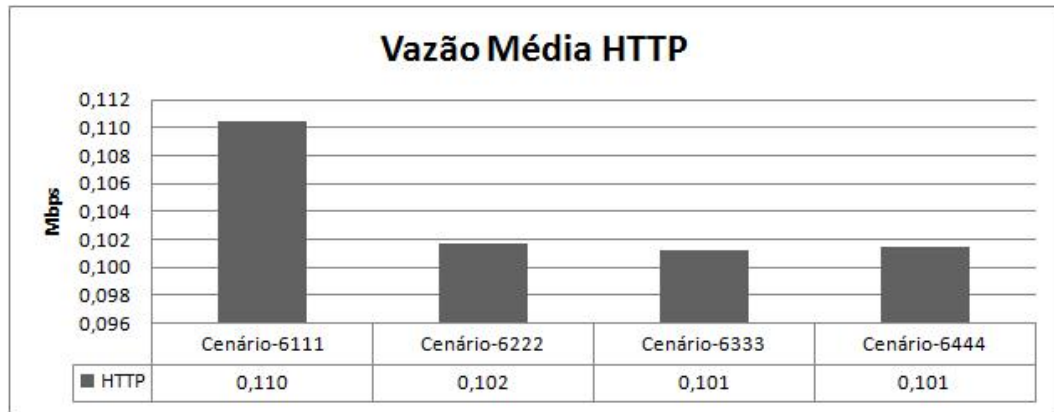


Figura 42: Vazão da aplicação HTTP para os diversos cenários.

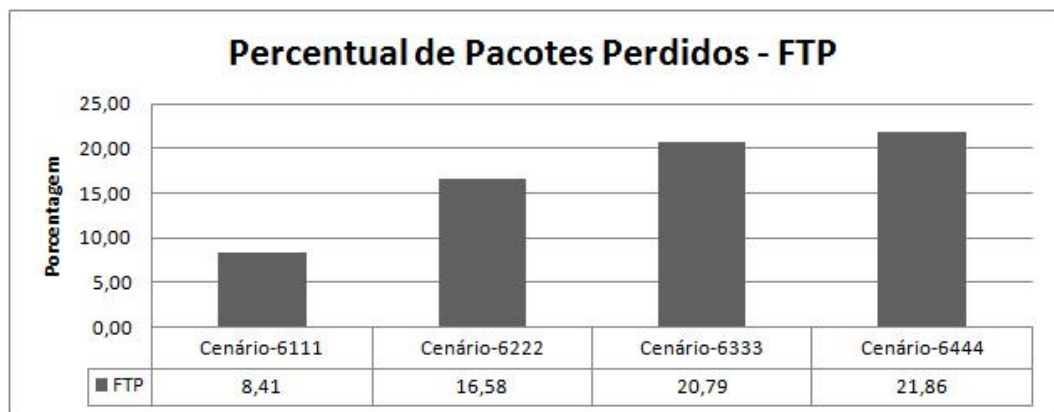


Figura 43: Percentual de perda de pacotes para a aplicação FTP para os diversos cenários.

Como se pode observar, há um impacto considerável nessa aplicação. A medida que novos *nós* são inseridos na rede, o percentual de perda de pacotes da aplicação aumenta. Para se ter uma idéia deste impacto, quando a aplicação FTP estava sendo executada sozinha catalogou-se uma perda de 5,84% de pacotes, já para o *Cenário-6111* verificou-se que este percentual subiu para 8,41%.

Por fim, para essas simulações verificou-se o impacto que os cenários propostos têm no jogo. O gráfico da Figura 44 mostra como se comportou o parâmetro de intervalo de envio e recebido do jogo.

Pode-se observar pelo gráfico da Figura 44, que, mesmo para o *Cenário-6444*,

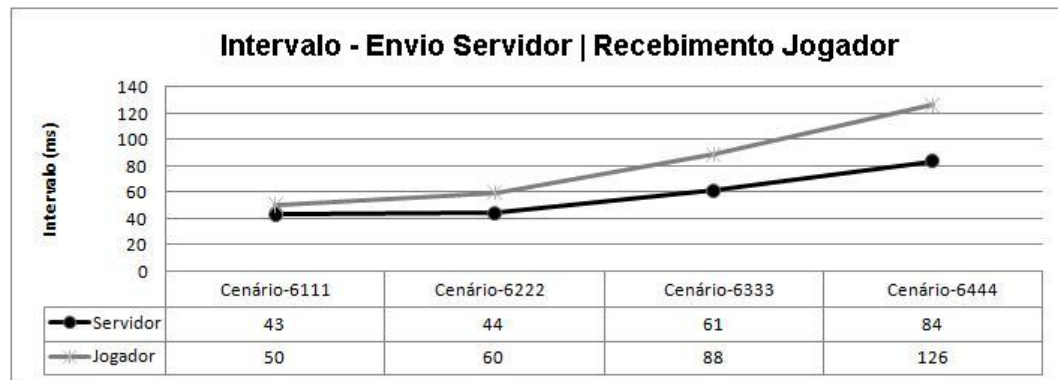


Figura 44: Intervalo de envio de pacotes do servidor e recebimento no jogador.

onde tínhamos 6 jogadores e 4 *nós* para cada aplicação avaliada, o intervalo de envio e recebimento de pacotes no jogo não atingiu o limite pré-estabelecido. O que nos indica que não houve prejuízo a jogabilidade para o jogo nos cenários avaliados, com apenas 6 jogadores.

6 CONCLUSÕES E TRABALHOS FUTUROS

6.1 Experimentos Reais

Este trabalho envolveu três etapas: experimentos reais, desenvolvimento de um modelo de simulação e avaliação de comportamento do jogo *OpenArena* em uma rede *Wi-Fi*, para cenários variados. Cada etapa teve suas contribuições. Os experimentos reais foram importantes para se conhecer os parâmetros da rede e posteriormente auxiliar na construção do modelo de simulação. Apesar de se especular que jogos degradam as redes de computadores, observou-se com poucos jogadores, que a utilização da largura de banda da rede não ultrapassou 2% de seu total. Já para cenários onde havia jogo e outras aplicações, observou-se que no último cenário do experimento 2, a utilização da rede chegou há 22.75% da capacidade total da rede.

Então, pode-se concluir que, a rede se comportou de maneira estável, não foi relatado nenhuma anomalia, degradação da jogabilidade (*lag*) ou queda de desempenho na utilização das aplicações, para os experimentos 1 e 2.

6.2 Modelo de Simulação

No modelo de simulação, foi possível, expandir os cenários reais, a fim de se identificar possíveis problemas na rede. Como contribuição deste trabalho, foi desenvolvido uma ferramenta que gera os *scripts* para serem executados no *NS2*. Esse programa foi desenvolvido utilizando-se a linguagem *java*. Conforme mencionado anteriormente, todos os parâmetros de entrada para as simulações foram extraídos das análises dos experimentos reais.

Um ponto importante deste trabalho foi o desenvolvimento de novos cenários de simulação, conforme apresentado no capítulo 5. Os cenários apresentados na seção 5.6, foram divididos em 4 fases. Aqui serão apresentadas as principais observações dessas fases. Para a fase 1, observou-se que apenas a aplicação FTP apresentou problemas, pois

a partir de três *nós* executando essa aplicação começa-se a perceber uma grande perda de desempenho na rede. Pode-se dizer pelos dados analisados que este tipo de aplicação teve um grande impacto na rede. Para a fase 2, foi verificado em quais pontos as aplicações selecionadas influenciam no desempenho do jogo. Pode-se observar que seis jogadores, um servidor e 15 aplicações *streaming* são suficientes para que os parâmetros de qualidade do jogo deixem de ser satisfeitos. Combinando o jogo com a aplicação *FTP* percebeu-se que acima de oito *nós* executado a aplicação a jogabilidade é degradada. Na fase 3 os pontos onde o jogo impacta nas aplicações foram mapeados. Nessa fase a aplicação que evidenciou um maior impacto foi o FTP, pode-se verificar que a cada novo jogador que ingressava na rede, a vazão média da aplicação diminuía e a perda de pacotes aumenta. Por fim, a fase 4, realizou um mistura entre as aplicações, esses cenários tendem a ser realísticos, pois em um ambiente real, temos vários tipos de aplicações em execução.

6.3 Trabalhos Futuros

Deve-se levar em consideração que existem outros tipos de tecnologia de rede e outros gêneros de jogos que precisam ser analisados. Sendo assim, faz-se necessário a construção de modelos que possam prever o comportamento de outros gêneros em outros ambientes. Como trabalhos futuros, pode-se criar outros modelos que utilizem técnicas analíticas ou de simulação.

Outros gêneros e outras tecnologias devem ser avaliados. Dentre eles, podemos citar:

1. Gêneros: MMORPG, Simulação, RTS entre outros
2. Comportamento em redes de celulares (3a geração, UMTS, etc.), WiMax, internet entre outras
3. Os consoles como Xbox, Wii, Nintendo DS, PSP2 e PSP3, entre outros

Realizar experimentos e simulações para jogos de realidade aumentada com *handoff* entre *AP's*

Existe uma infinidade de possibilidades quando se estuda o comportamento de jogos em rede, e o grande desafio é conhecer e modelar o comportamento destes aplicativos nas novas tecnologias de rede que estão surgindo, para que seja possível a convivência de usuários de entretenimento com os demais, sem prejuízo de desempenho.

REFERÊNCIAS

3GPP. 2009. Acessado em Julho de 2009. Disponível em: <<http://www.3gpp.org/>>.

ABRAGAMES. Novembro 2009. Acessado em 20/11/2009. Disponível em: <www.abragames.org/docs/Abragames-Pesquisa2008.pdf>.

AZEVEDO, E. *Desenvolvimento de Jogos 3D e Aplicações em Realidade Virtual*. [S.l.]: Elsevier, 2005.

BRANCH PHILIP; ARMITAGE, G. 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, 2006. Braunschweig, Germany.

BRUN JEREMY; SAFAEI, F. B. P. Server topology considerations in online games. *ACM Press. Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, 2006.

CARRIG BRIAN; DEFIEFFE, D. M. J. 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 2007. Disponível em: <<http://ieeexplore.ieee.org/>>.

CLAYPOOL, M. On the 802.11 turbulence of nintendo ds and sony psp hand-held network games. *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*, ISBN:1-59593-156-2, 2005. Hawthorne, NY.

FARBER, J. Network game traffic modelling. *Proceedings of the 1st workshop on Network and system support for games*, ISBN:1-58113-493-2, 2002. Braunschweig, Germany.

IEEE-STANDARD. 2010. Acessado em Junho 2010. Disponível em: <<http://standards.ieee.org/getieee802/802.11.html>>.

INFO-REVIEWS. 2010. Acessado em Fevereiro de 2010. Disponível em: <<http://info.abril.com.br/reviews/hardware/redes/seguranca-e-o-forte-do-wrt54g.shtml>>.

KINICK JAMES; CLAYPOOL, M. Traffic analysis of avatars in second life. *ACM Proceedings - International Workshop on Network and Operating System Support for Digital Audio and Video.*, 2008.

LANG TANJA; BRANCH, P. A. G. A synthetic traffic model for quake3. *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, ISBN:1-58113-882-2, 2004. Singapore.

LV SHAOHE; WANG, X. Z. X. Applying active measurement in the rate adaption for 802.11 wireless network. *ACM International Conference Proceedings Series*, 2008. ISBN: 978-963-9799-36-3.

MILLINGTON, I. *Artificial Intelligence for Games*. [S.l.]: Elsevier, 2006.

NPDGROUP. Novembro 2009. Acessado em 20/11/2009. Disponível em: <www.npd.com>.

NS2. 2009. Acessado em Novembro de 2009. Disponível em: <<http://www.isi.edu/nsnam/ns/>>.

NS2-WIKI. 2009. Acessado em Julho de 2009. Disponível em: <<http://en.wikipedia.org/wiki/Ns2>>.

OLIVER IAIN; MILLER, A. A. C. Virtual worlds, real traffic: Interaction and adaptation. *ACM Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, 2010. ISBN: 978-1-60558-914-5.

OPENARENA-FTP. 2009. Acessado em novembro de 2009. Disponível em: <<http://www.openarena.ws/files.html>>.

OPENARENA-WIKI. 2009. Acessado em Junho de 2009. Disponível em: <<http://openarena.wikia.com>>.

RUTTER, J. B. J. *Understanding Digital Games*. [S.l.]: Longon : Sage, 2006, 2006.

S RATTI; B, H. S. S. A survey and analysis of first person shooter gaming traffic on the internet. *Internet Computing, IEEE*, 2010. ISBN: 978-963-9799-36-3.

WIRESHARK. 2009. Acessado em Novembro de 2009. Disponível em: <<http://www.wireshark.org/docs/>>.

ZANDER, S. A traffic model for the xbox game halo 2.... *Proceedings of the international workshop on Network and operating systems support for digital audio and video table of contents*, ISBN:1-58113-987-X, 2005. Washington, USA.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)