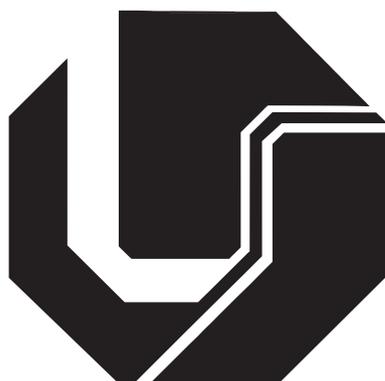


**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**UMA PROPOSTA DE PLATAFORMA DE SIMULAÇÃO PARA ESTUDO
DO PROTOCOLO ENUM**

Saulo Henrique da Mata

Uberlândia - 2012

Saulo Henrique da Mata

**UMA PROPOSTA DE PLATAFORMA DE SIMULAÇÃO PARA ESTUDO
DO PROTOCOLO ENUM**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do grau de Mestre em Ciências, aprovada em 10 de Fevereiro de 2012 pela banca examinadora:

Paulo Roberto Guardieiro, Dr. - Orientador (UFU)

Ruy de Oliveira, Dr. (IFMT)

Éderson Rosa da Silva, Dr. (UFU)

Saulo Henrique da Mata

**UMA PROPOSTA DE PLATAFORMA DE SIMULAÇÃO PARA ESTUDO
DO PROTOCOLO ENUM**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do grau de Mestre em Ciências.

Prof. Paulo Roberto Guardieiro, Dr.
Orientador

Prof. Alexandre Cardoso, Dr.
Coordenador do Programa de Pós-Graduação

Agradecimentos

Agradeço primeiramente a Deus por estar comigo a cada dia. Agradeço pela perseverança, saúde e alegria que estiveram presentes durante todos os dias de trabalho.

Ao prof. Dr. Paulo Roberto Guardieiro pela sempre dedicada e paciente orientação. Agradeço pelo incentivo e por me ensinar não somente conceitos técnicos, mas também por ser exemplo de conduta e profissionalismo.

À minha família pelo apoio em todos os momentos. Agradeço à minha mãe Domingas, ao meu pai José e à minha namorada Joyce pelo apoio nos momentos difíceis, por compreenderem a minha ausência em tantos momentos e pela confiança depositada em mim.

Agradeço aos meus amigos de república e do trabalho pela companhia, apoio, aprendizado, e tantos momentos de descontração e alegria.

Aos professores e técnicos da Faculdade de Engenharia Elétrica pelo aprendizado e colaboração.

Por fim, agradeço a CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) e a FINEP (Financiadora de Estudos e Projetos) pelo auxílio financeiro para a viabilização deste trabalho.

"Tudo posso naquele que me fortalece."

Fl 4,13

Resumo

da Mata, S. H., *UMA PROPOSTA DE PLATAFORMA DE SIMULAÇÃO PARA ESTUDO DO PROTOCOLO ENUM*, UFU, Uberlândia, Brasil, 2012, 106p.

As redes IP têm se tornado uma opção cada vez mais vantajosa em relação às redes de comunicações tradicionais. Mesmo diante das vantagens oferecidas pela tecnologia VoIP em relação à PSTN, não é possível uma substituição instantânea de uma tecnologia pela outra devido à complexidade envolvida. Isto implica em um período de transição em que as duas tecnologias irão conviver. Para que isto seja realizável, é necessário que exista um sistema de mapeamento entre números E.164 (utilizados na PSTN) e URIs (utilizados na rede IP). É neste contexto que o protocolo ENUM vem ganhando destaque como a solução mais promissora. Entretanto, a sua disseminação ainda é baixa. Além disso, existem questões técnicas e, principalmente, interesses políticos e econômicos que precisam ser trabalhados para que este protocolo seja popularizado. Em relação às questões técnicas, existe uma escassez de ferramentas para o estudo deste protocolo. Neste sentido, esta dissertação apresenta uma proposta de plataforma de simulação que possibilite o estudo do protocolo ENUM. Dessa maneira, o projeto de concepção do simulador foi baseado na realização de testes em laboratório, na proposição de modelos matemáticos e, posteriormente, na implementação, integração e validação do simulador. Os resultados indicam que a plataforma de simulação descreve com fidelidade o sistema real considerado. Os resultados também comprovaram o sucesso da estratégia de modelagem proposta. Além disso, novos módulos foram desenvolvidos para o ambiente de simulação NS-3, oferecendo novas possibilidades para os pesquisadores da área.

Palavras-chave: ENUM, DNS, Modelagem, Simulação, NS-3, Estudo de Desempenho

Abstract

da Mata, S. H., *A PROPOSAL OF SIMULATION PLATFORM FOR STUDY OF ENUM PROTOCOL* , UFU, Uberlândia, Brazil, 2012, 106p.

IP networks have become an increasingly profitable option over traditional communication networks. Despite the advantages of VoIP compared to PSTN, it is not possible to replace one technology by the other instantly, due to the complexity involved. This implies a transition period in which the two technologies will coexist. In order to make this feasible, there must be a system for mapping E.164 numbers (used in the PSTN) to URIs (used in the IP network). In this context, the ENUM protocol is gaining prominence as the most promising solution. However, it is not very spread yet. In addition, there are technical issues and especially political and economic interests that need to be worked so that this protocol can be popularized. Regarding technical issues, there is a shortage of tools for the study of this protocol. Thus, this dissertation proposes a simulation platform that enables the study of the ENUM protocol. The project design of the simulator was based on experiments carried out in the laboratory, the proposition of mathematical models and, subsequently, implementation, integration and validation of the simulator. The results indicate that the simulation platform rightly describes the real system considered. The results also proved the success of the proposed modeling strategy. In addition, new modules were developed for the NS-3 simulation environment, offering new opportunities for researchers.

Index-terms: ENUM, DNS, Modeling, Simulation, NS-3, Study of Performance

Sumário

Sumário	x
Lista de Figuras	xiii
Lista de Tabelas	xiv
Lista de Abreviaturas e Siglas	xvii
1 Introdução	1
1.1 Definição do Problema e Solução Proposta	2
1.2 Trabalhos Relacionados	4
1.3 Justificativa	6
1.4 Objetivos e Metas	7
1.5 Estrutura da Dissertação	7
2 O Protocolo ENUM	9
2.1 Introdução	9
2.2 <i>Domain Name System (DNS)</i>	10
2.2.1 Espaço de Nomes do DNS	11
2.2.2 Zonas	11
2.2.3 <i>Resolver</i>	12
2.2.4 Registros de Recursos	12
2.2.5 Servidores de Nomes	13
2.2.6 Funcionamento do DNS	13
2.3 Números telefônicos E.164	15
2.4 <i>Uniform Resource Identifiers (URIs)</i>	16
2.5 <i>Application Unique String (AUS)</i>	16

2.6	Registro <i>Name Authority Pointer</i> (NAPTR)	17
2.7	<i>Session Initiation Protocol</i> (SIP)	18
2.8	Descrição Geral do Protocolo ENUM	18
2.8.1	O protocolo ENUM no contexto PSTN-VoIP	18
2.8.2	Aplicações ENUM e o DDDS	21
2.8.3	Arquitetura ENUM	21
2.8.4	Tipos de ENUM	22
2.8.5	Funcionamento do ENUM	23
2.9	Considerações Finais	24
3	Uma Proposta de Plataforma de Simulação para o Estudo do Protocolo ENUM	26
3.1	Introdução	26
3.2	Descrição do Problema	26
3.3	Descrição da Plataforma de Simulação	28
3.3.1	Considerações Iniciais e Simulador Protótipo	28
3.3.2	Definição do Sistema Real	31
3.3.3	Ambiente de Implementação do Simulador	32
3.4	Considerações Finais	34
4	Descrição dos Testes em Laboratório	35
4.1	Introdução	35
4.2	Perfil do Hardware das Máquinas	35
4.3	Softwares	36
4.3.1	Sistema Operacional	36
4.3.2	Software DNS/ENUM	36
4.3.3	<i>EnumBenchTool</i>	37
4.3.3.1	<i>dnsperf</i>	37
4.3.3.2	<i>top</i>	38
4.3.3.3	<i>bwm-ng</i>	38
4.3.3.4	Configuração de Sessão de Testes	38
4.4	Metodologia de Teste	39
4.4.1	Layout do Ambiente de Teste	39

4.4.2	Tamanho do Banco de Dados	39
4.4.3	Arquivos de Consultas	40
4.4.4	Arquivos para Zonas DNS (<i>Zone Files</i>)	40
4.5	Resultados dos Testes em Laboratório	41
4.6	Considerações Finais	43
5	Descrição do Modelo de Simulação	48
5.1	Introdução	48
5.2	Modelo para a ferramenta <i>dnstperf</i>	48
5.3	Modelagem dos Servidores Autoritativos	49
5.3.1	Considerações sobre a Modelagem	49
5.3.2	Modelagem Baseada em Regressão de Funções	50
5.3.2.1	Regressão Linear	50
5.3.2.2	Regressão Exponencial	51
5.3.2.3	Qualidade da Regressão	52
5.3.2.4	Identificação de <i>Outliers</i>	53
5.3.2.5	Equações de Predição	55
5.3.3	Modelagem Direta Baseada em Amostras	57
5.4	Modelagem do Enlace de Comunicação	61
5.5	Considerações Finais	62
6	Implementação da Plataforma de Simulação	63
6.1	Introdução	63
6.2	Considerações de Projeto do Simulador	63
6.2.1	Ambiente de Implementação	64
6.2.2	Módulos Desenvolvidos	65
6.2.3	Arquivos de Configuração	66
6.3	Implementação do Módulo <i>DnsQueryTool</i>	68
6.4	Implementação do Módulo <i>DnsAuthoritativeServer</i>	70
6.5	Integração do Simulador ao NS-3	71
6.5.1	Classe <i>DnsHeader</i>	72
6.5.2	Classe <i>DnsPacketTag</i>	74
6.5.3	Classe <i>EnumTimeDataCalculator</i>	75

6.5.4	Classe <i>ThroughputCalculator</i>	75
6.5.5	Classe <i>DnsQueryToolHelper</i>	75
6.5.6	Classe <i>DnsAuthoritativeServerHelper</i>	75
6.6	Módulo <i>DnsConfigTool</i>	75
6.6.1	Classes <i>FileParser</i> , <i>UnityParser</i> e <i>SetupTool</i>	76
6.6.2	Classes <i>QueryFileGenerator</i> e <i>ZoneFilesGenerator</i>	76
6.7	Validação da Plataforma de Simulação	76
6.8	Considerações Finais	80
7	Conclusões Gerais	86
	Referências Bibliográficas	89
A	Dados dos Testes em Laboratório	95
B	Conceitos Matemáticos	100
B.1	O Método dos Mínimos Quadrados	100
B.2	Teste <i>z-score</i> Modificado	101
B.3	Coefficiente de Determinação	102
B.4	Coefficiente de Correlação	102
B.5	Intervalo de Confiança para β_0 e β_1	102
B.6	Intervalo de Predição	103
C	Dados dos Testes de Simulação	104

Lista de Figuras

2.1	O ENUM possibilita a conexão de vários serviços com apenas um único número telefônico E.164 [11].	10
2.2	Estrutura hierárquica do DNS.	12
2.3	Funcionamento típico do DNS.	14
2.4	Estrutura dos números telefônicos E.164 para áreas geográficas [27].	16
2.5	Interconexão de Operadoras VoIP por meio da PSTN.	18
2.6	Interconexão de Operadoras VoIP por meio de uma rede IP.	20
2.7	ENUM: exemplo de resolução de um número telefônico E.164.	24
3.1	Topologia de testes utilizada no protótipo da plataforma de simulação [33].	28
3.2	Diagrama simples de um cenário de consultas do protocolo ENUM.	31
3.3	Diagrama simples do sistema real avaliado em laboratório.	32
4.1	Layout do ambiente de testes de avaliação de desempenho de um servidor DNS/ENUM.	39
4.2	Desempenho do servidor DNS/ENUM com relação a vazão de respostas às consultas.	43
4.3	Comportamento do tempo médio de resposta.	44
4.4	Comportamento da taxa de perdas.	44
4.5	Comportamento do desvio padrão do tempo médio de resposta.	45
4.6	Utilização da CPU pelo software BIND.	45
4.7	Utilização da CPU pela <i>dnstperf</i>	46
4.8	Utilização do enlace de comunicação.	46
5.1	Comportamento do tempo médio de resposta (1000 amostras).	53
5.2	Comportamento do desvio padrão do tempo médio de resposta (1000 amostras).	54

5.3	Comportamento da taxa de perdas (1000 amostras).	54
5.4	Comportamento do tempo médio de resposta (<i>outliers</i> removidos).	55
5.5	Comportamento do desvio padrão do tempo médio de resposta (<i>outliers</i> removidos).	56
5.6	Comportamento da taxa de perdas (<i>outliers</i> removidos).	56
5.7	Comparação entre comportamento real e o modelo proposto para o tempo médio de resposta, considerando-se 500k registros no banco de dados.	59
5.8	Comparação entre comportamento real e o modelo proposto para o desvio padrão do tempo médio de resposta, considerando-se 500k registros no banco de dados.	60
5.9	Comparação entre comportamento real e o modelo proposto para a taxa de perdas, considerando-se 500k registros no banco de dados.	61
6.1	Organização do NS-3 [49].	65
6.2	Topologia do Simulador.	66
6.3	Arquivos de configuração requeridos por cada módulo da plataforma de simulação.	69
6.4	Diagrama de fluxo para o módulo <i>DnsQueryTool</i>	70
6.5	Diagrama de fluxo para o módulo <i>DnsAuthoritativeServer</i>	71
6.6	Integração da plataforma de simulação ao NS-3.	72
6.7	Cabeçalho DNS no sistema real.	73
6.8	Cabeçalho DNS no ambiente de simulação.	74
6.9	Desempenho do servidor DNS/ENUM baseado em simulação com relação a vazão de respostas às consultas (modelado por regressão de funções).	77
6.10	Comportamento do tempo médio de resposta baseado em simulação (modelado por regressão de funções).	78
6.11	Comportamento do desvio padrão do tempo médio de resposta baseado em simulação (modelado por regressão de funções).	79
6.12	Comportamento da taxa de perdas baseada em simulação (modelado por regressão de funções).	80
6.13	Desempenho do servidor DNS/ENUM baseado em simulação com relação a vazão de respostas às consultas (modelado por amostras).	81

6.14	Comportamento do tempo médio de resposta baseado em simulação (modelado por amostras).	81
6.15	Comportamento do desvio padrão do tempo médio de resposta baseado em simulação (modelado por amostras).	82
6.16	Comportamento da taxa de perdas baseado em simulação (modelado por amostras).	82
6.17	Comparação do erro cometido por cada estratégia de modelagem com relação a vazão de respostas às consultas.	83
6.18	Comparação do erro cometido por cada estratégia de modelagem com relação ao tempo médio de resposta.	83
6.19	Comparação do erro cometido por cada estratégia de modelagem com relação ao desvio padrão do tempo médio de resposta.	84
6.20	Intervalo de confiança (95%) para a vazão de respostas às consultas, considerando-se o modelo baseado na utilização direta de amostras.	84
6.21	Intervalo de confiança (95%) para a vazão de respostas às consultas, considerando-se o modelo baseado na utilização direta de amostras.	85

Lista de Tabelas

3.1	Comparação do desempenho dos simuladores proposto em [24] e [33]. . . .	29
4.1	Intervalos de FQDNs para os arquivos de consultas.	40
4.2	Zonas para cada grupo de registros.	40
5.1	Equações de predição para o servidor DNS/ENUM considerando-se 500k registros.	58
A.1	Dados dos testes realizados em laboratório (500k registros)	96
A.2	Dados dos testes realizados em laboratório (5M registros)	98
C.1	Dados dos testes de simulação, considerando-se o modelo baseado na regressão de funções (500k registros)	104
C.2	Dados dos testes de simulação, considerando-se o modelo baseado na utilização direta de amostras (500k registros)	105

Lista de Abreviaturas e Siglas

AUS	<i>Application Unique String</i>
BIND	<i>Berkeley Internet Name Domain</i>
BSD	<i>Berkeley Software Distribution</i>
BWM-NG	<i>Bandwidth Monitor NG</i>
CGI.br	Comitê Gestor da Internet no Brasil
CND	Código Nacional de Destino
CPS	Consultas Por Segundo
CPU	<i>Central Processing Unit</i>
DDD	Discagem Direta à Distância
DDDS	<i>Dynamic Delegation Discovery System</i>
DLZ	<i>Dynamically Loadable Zones</i>
DNS	<i>Domain Name System</i>
DNSSEC	<i>Domain Name System Security Extensions</i>
ENUM	<i>Electronic Numbering</i>
FQDN	<i>Fully Qualified Domain Name</i>
FTP	<i>File Transfer Protocol</i>
IP	<i>Internet Protocol</i>
ISC	<i>Internet Software Consortium</i>

ITU-T	<i>International Communication Union - Telecommunication Standardization Sector</i>
LDAP	<i>Lightweight Directory Access Protocol</i>
MAD	<i>Median Absolute Deviation</i>
MGCF	<i>Media Gateway Control Function</i>
MGW	<i>Media Gateway</i>
MQE	Média dos Quadrados dos Erros
NAPTR	<i>Name Authority Pointer</i>
NGN	<i>Next Generation Network</i>
NS-2	<i>Network Simulator 2</i>
NS-3	<i>Network Simulator 3</i>
ODBC	<i>Open Database Connectivity</i>
OTcl	<i>Object-oriented Tool Command Language</i>
PLMN	<i>Public Land Mobile Network</i>
PSTN	<i>Public Switched Telephone Network</i>
RFC	<i>Request For Comments</i>
RIPE-NCC	<i>Réseaux IP Européens Network Coordination Centre</i>
RR	<i>Resource Record</i>
SGW	<i>Signalling Gateway</i>
SIP	<i>Session Initiation Protocol</i>
SLD	<i>Second Level Domain</i>
SMS	<i>Short Message Service</i>
SQE	Soma dos Quadrados dos Erros

TLD	<i>Top Level Domain</i>
TSIG	<i>Transaction Signature</i>
TTL	<i>Time To Live</i>
TU-COM	Tarifa de Uso de Comutação
UDP	<i>User Datagram Protocol</i>
URI	<i>Uniform Resource Identifier</i>
VoIP	<i>Voice over IP</i>

Capítulo 1

Introdução

O avanço tecnológico proporcionou à sociedade moderna um leque variado de opções quando o assunto é a comunicação. Dentre as diversas possibilidades existentes é possível citar:

- Voz sobre a *Public Switched Telephone Network* (PSTN);
- Voz sobre a *Public Land Mobile Network* (PLMN);
- Voz sobre a rede IP (*Voice over IP* - VoIP);
- mensagens de texto (*Short Message Service* - SMS);
- correio eletrônico;
- fotos e vídeos.

Tem se tornado comum a visão de que futuramente todas as redes de comunicações serão baseadas na rede IP (*Internet Protocol*). Esta tecnologia permite a integração de voz, dados e outros tipos de serviços em uma única rede por um custo menor que o praticado hoje pelas tradicionais redes de comunicações [1]. Não obstante às vantagens oferecidas pela tecnologia VoIP em relação à PSTN, não é possível uma substituição instantânea de uma tecnologia pela outra devido à complexidade e custos envolvidos. Neste sentido, existirá um período de transição em que as duas tecnologias irão conviver, uma vez que as operadoras de telecomunicações realizaram grandes investimentos na PSTN, que podem ainda não ter sido completamente amortizados, e a migração dos assinantes da PSTN

para o serviço VoIP pode ocorrer de forma lenta e gradual, de acordo com o interesse dos usuários, operadoras e agências reguladoras.

Considerando-se as tecnologias PSTN e VoIP, é possível encontrar quatro cenários básicos de chamadas telefônicas: PSTN-PSTN, PSTN-IP, IP-PSTN e IP-IP [1]. Tradicionalmente, números telefônicos E.164 e *Uniform Resource Identifiers* (URIs) são utilizados na PSTN e na Internet, respectivamente, para a identificação de usuários. Protocolos de sinalização como o *Session Initiation Protocol* (SIP) e o H.323 podem trabalhar com estes dois tipos de identificadores, além de endereços IP. Dessa maneira, para usuários na Internet, que geralmente possuem um teclado alfanumérico disponível, nomes do tipo textual (URIs) são mais utilizados devido à facilidade de memorização e dedução. Contudo, um problema surge quando as redes VoIP e PSTN são interconectadas, uma vez que os terminais da PSTN, tipicamente, oferecem aos usuários um teclado numérico. Esta realidade traz à tona o advento da *Next Generation Network* (NGN) que oferece uma via de migração das redes tradicionais para a Internet, enquanto mantém os serviços existentes [2].

Para que o cenário descrito anteriormente seja factível, é necessário que exista um sistema de mapeamento entre números telefônicos E.164 e URIs que possibilite a interconexão, por exemplo, das redes VoIP e PSTN. Esse sistema de endereçamento e nomes deve utilizar os padrões já existentes, uma vez que a implementação de novos protocolos ou alterações substanciais na arquitetura atual podem ser impraticáveis devido ao tamanho e a complexidade das redes de comunicações atuais. É nesse contexto que a utilização do *Telephone Number Mapping* ou *Electronic Numbering* (ENUM) se destaca como uma solução plausível.

1.1 Definição do Problema e Solução Proposta

O ENUM é um protocolo de mapeamento de números E.164 utilizados em redes PSTN para URIs utilizados na Internet. Este protocolo facilita a convergência dos serviços de Internet com os serviços das redes de telecomunicações existentes [3] utilizando a estrutura do *Domain Name System* (DNS) já consolidada pelo mundo.

Embora o ENUM seja uma tecnologia relativamente madura, integradora e que, potencialmente, reduzirá os custos de operação das redes de telecomunicações [4], nota-se

que sua implantação ocorre de forma gradual e de acordo com o interesse de cada país. Por outro lado, existem muitos desafios para o ENUM se consolidar como protocolo de interconexão entre as redes de comunicações existentes, principalmente devido às questões de regulamentação e modelo de negócios para as operadoras. Além disso, existem poucas experiências de implantação real deste protocolo e questões de ordem técnica também devem ser consideradas, as quais demandam estudos sobre o funcionamento e desempenho do protocolo.

Para mapear números telefônicos E.164 em URIs, o protocolo ENUM utiliza um tipo especial de registro de recurso (*Resource Record* - RR) chamado *Name Authority Pointer* (NAPTR) [5]. Neste contexto, apesar de utilizar o DNS como banco de dados, o ENUM possui alguns requisitos mais restritivos que o DNS clássico e, conseqüentemente, sua inserção ao DNS causaria alguns impactos na estrutura existente. Dentre estes fatores destaca-se:

- Tempo de estabelecimento da chamada: Para oferecer um serviço com qualidade similar à oferecida pela PSTN, o tempo de estabelecimento de uma chamada VoIP não deve ultrapassar o tempo que a PSTN gasta para completar o estabelecimento de uma chamada.
- Atualização dos registros: As informações armazenadas no DNS que são acessadas pela aplicação ENUM irão mudar frequentemente. Mesmo atualmente, as mudanças dos dados no DNS têm de estar globalmente disponíveis quase que instantaneamente devido a questões de regulamentação do serviço e, portanto, é importante que o servidor DNS/ENUM seja capaz de fornecer altas taxas de atualização de registros sem a interrupção do serviço [3].
- Tamanho da resposta: Registros NAPTR para o ENUM podem ser significativamente maiores que registros DNS típicos. Um número E.164 pode gerar de 10 a 20 registros NAPTR [2]. Tais mensagens de resposta poderão exceder o tamanho típico do segmento UDP utilizado na Internet (512 bytes), dificultando a utilização do UDP como protocolo de transporte, além do aumento do tráfego na rede.
- Mecanismo de *Cache*: A distribuição dos números telefônicos E.164 em alguns casos pode ser diferente da atual distribuição de nomes de domínios e o mecanismo de

cache, que é um fator crítico para a escalabilidade do DNS [6], pode não ser eficiente o bastante.

- Privacidade e Segurança: A utilização do ENUM pode implicar na publicação de informações pessoais (endereço, telefone, e-mail, etc) na Internet. É de fundamental importância que dispositivos de segurança assegurem o sigilo dos dados.

Estes são alguns exemplos de fatores de impacto da implementação do protocolo ENUM sobre a base de dados DNS existente. Dessa maneira, o conhecimento dos parâmetros que influenciam estes fatores é de fundamental importância para uma implementação bem sucedida desta tecnologia. O problema reside, portanto, no desenvolvimento de metodologias para avaliar e acompanhar o comportamento destes parâmetros diante de diferentes cenários de rede. A realização de medições em um sistema real para efetuar tais avaliações, é uma primeira opção a ser considerada. Entretanto, nem sempre é possível aplicar tal metodologia, devido à falta de equipamentos, custos envolvidos e, até mesmo, a não existência de uma rede implementada, como, por exemplo, em estudos para implantação inicial do serviço. Neste contexto, o estudo dos parâmetros por meio de modelagem e simulação surge como uma solução viável e confiável. Em um ambiente de simulação é possível construir diferentes cenários, e controlar os parâmetros de acordo com o objetivo do estudo e encontrar resultados próximos aos de um sistema real por um custo bem menor. Diante do exposto, esta dissertação apresenta uma proposta para solucionar o problema do estudo dos principais parâmetros de influência em uma implementação do protocolo ENUM. A proposta é baseada no desenvolvimento de uma plataforma de simulação que possibilite ao usuário a construção de um ambiente onde é possível controlar e avaliar tais parâmetros.

1.2 Trabalhos Relacionados

Esta seção tem por objetivo levantar os principais trabalhos desenvolvidos pela comunidade acadêmica sobre o protocolo ENUM e relacioná-los com a proposta aqui apresentada. Dentre estes trabalhos, é possível identificar, no geral, enfoques nas áreas de estudos para implantação do serviço, modelos de negócios e modelos administrativos, *benchmarking* de servidores DNS/ENUM, bem como estudo de desempenho por meio de modelagem e simulação.

Em [7] é possível encontrar um estudo sobre as potencialidades do ENUM em seus primeiros anos de criação. Modelos de administração, oportunidades de mercado e a relação entre as diferentes entidades diretamente ligadas ao ENUM são alguns dos temas abordados neste trabalho.

No que concerne aos estudos para implantação do serviço, nota-se que este processo acontece de forma gradual, de acordo com as iniciativas e o interesse de cada país. Estes estudos são geralmente baseados nas possibilidades existentes para efetuar esta implantação, isto é, na identificação das opções quanto ao tipo do serviço (*User ENUM*, *Infrastructure ENUM*, *Carrier ENUM*, etc), tipo de arquitetura, modelo de negócios, questões de privacidade e segurança, legislação, etc. Por outro lado, tais trabalhos não analisam questões mais específicas do protocolo ENUM ou dos servidores DNS/ENUM como, por exemplo, o tempo de resposta para resolução de números E.164 em um URI, escalabilidade, vazão dos servidores, etc. Neste contexto, [8] apresenta uma visão geral da implantação do serviço ENUM em alguns países da Europa. Além do mais, pode-se identificar outros trabalhos direcionados à implantação do ENUM em determinados países, nomeadamente, Austrália ([9], [10] e [11]), Áustria ([12], [13]), Holanda ([14] e [15]), Irlanda ([16]), Japão ([17]), Malásia ([18]), Portugal ([4]), dentre outros.

No que tange ao *benchmarking* de servidores DNS/ENUM, as principais referências são os trabalhos desenvolvidos em [19] e [20]. Estes trabalhos apresentam um estudo comparativo do desempenho dos softwares BIND, PowerDNS e Navitas com relação a algumas métricas como vazão, tempo de resposta, escalabilidade, atualização do banco de dados, dentre outros. Segundo os autores, os softwares Navitas e PowerDNS cumprem os requisitos exigidos pelo ENUM, sendo o Navitas o software com melhor desempenho. O software BIND, por outro lado, não foi qualificado como um software adequado para a utilização em um contexto ENUM.

O trabalho apresentado em [21] realiza um estudo da portabilidade numérica e avalia o sistema de atualização de registros de um sistema ENUM. Utilizando o método “*DNS zone transfer*” para sincronização de banco de dados, são avaliados dois softwares para servidores DNS/ENUM: BIND e PowerDNS. Neste estudo de *benchmarking*, a métrica considerada foi a velocidade na transferência de informações entre diferentes zonas. Neste quesito os resultados mostraram que o software BIND entregou os dados de portabilidade numérica com um desempenho 3 a 4 vezes melhor que o software PowerDNS.

A área de estudo do protocolo ENUM por meio de modelagem e simulação é basicamente representada pelos trabalhos apresentados em [22], [23] e [24]. Nestes trabalhos é possível encontrar o desenvolvimento de um módulo de simulação para o Simulador *Network Simulator 2* (NS-2). A modelagem utilizada baseou-se em medições realizadas no sistema ENUM francês. Tais referências apresentam apenas o processo de construção do módulo de simulação (metodologia de testes, modelagem e implementação) e não apresentam resultados obtidos a partir do módulo simulador implementado. Maiores detalhes sobre o módulo de simulação e alguns resultados de simulação podem ser encontrados em [2].

1.3 Justificativa

A partir dos trabalhos citados na seção anterior é possível observar que existe uma carência de ferramentas que possibilitem o estudo das principais variáveis relacionadas ao protocolo ENUM.

Apesar do trabalho apresentado em [24] desenvolver uma ferramenta com este objetivo, nota-se, ao analisar-se os resultados apresentados em [2], que o módulo de simulação possui algumas deficiências, dentre elas:

- o estudo apresentado considera apenas uma estreita faixa de valores para os parâmetros de simulação;
- o módulo não apresenta flexibilidade quanto a inserção de novos tipos de arquitetura e softwares DNS/ENUM;
- o módulo foi criado para ser acoplado ao simulador NS-2 que atualmente recebe poucas atualizações e tende a ficar obsoleto.

Considerando, portanto, que existem poucos trabalhos na área do estudo do protocolo ENUM por meio de modelagem e simulação e considerando ainda que as propostas existentes possuem algumas deficiências, a proposta apresentada nesta dissertação almeja contribuir para que novas ferramentas de estudo sejam disponibilizadas.

1.4 Objetivos e Metas

O objetivo geral desta dissertação é propor e desenvolver uma ferramenta de simulação que possibilite o estudo do protocolo ENUM. Para alcançar este objetivo foram definidas algumas metas:

- Definição do modelo a ser utilizado para abstração do sistema real;
- Definição do ambiente de implementação da ferramenta de simulação;
- Execução de testes em laboratório para a aquisição de dados do sistema real;
- Validação da ferramenta de simulação a partir da comparação dos resultados de simulação com os resultados dos testes em laboratório;

Espera-se assim, que a ferramenta de simulação proposta contribua para o desenvolvimento do protocolo ENUM, facilitando o seu estudo por parte da academia, bem como auxiliando a implantação de aplicações baseadas neste protocolo.

1.5 Estrutura da Dissertação

Com o intuito de atender as metas anteriormente definidas, esta dissertação apresenta-se estruturada, além deste capítulo introdutório, com as unidades descritas a seguir.

O Capítulo 2 é dedicado à apresentação de conceitos e fundamentos que facilitam o entendimento do protocolo ENUM. Esse capítulo também é responsável por apresentar o próprio protocolo e seu contexto de atuação.

No Capítulo 3 são apresentadas as linhas gerais da plataforma de simulação proposta, com a definição do problema, descrição do sistema real considerado e apresentação do ambiente de implementação.

O Capítulo 4 é dedicado à descrição dos testes realizados em laboratório. Nesse capítulo são apresentados os procedimentos de testes, os equipamentos utilizados e os resultados obtidos em laboratório.

O Capítulo 5 apresenta os modelos matemáticos propostos para descrever o comportamento do sistema real analisado nos testes em laboratório.

O Capítulo 6 é dedicado à apresentação do processo de implementação, integração e validação da plataforma de simulação proposta. Nesse capítulo são apresentados os resultados de simulação obtidos a partir do modelo de simulação.

O Capítulo 7 finaliza esta dissertação apresentando as conclusões, considerações finais e as oportunidades para eventuais trabalhos futuros.

Capítulo 2

O Protocolo ENUM

2.1 Introdução

O ENUM é um protocolo e um banco de dados que mapeia um número de telefone E.164 em um URI. Percebe-se que tal protocolo se apresenta como uma solução para o acesso de serviços na Internet utilizando apenas um número telefônico E.164. Além disso, o ENUM tem o potencial de facilitar o desenvolvimento de aplicações e serviços como o VoIP, aplicações multimídia, vídeo conferência, mensagens instantâneas dentre outros.

Por um lado, ao ser usado a partir de dispositivos telefônicos tradicionais, integrados numa rede VoIP compatível com ENUM, a chamada poderá ser entregue, contornando as redes tradicionais de voz e, conseqüentemente, reduzindo os custos de chamada, sem perda de qualidade [4].

Por outro lado, ao ser utilizado com dispositivos mais modernos (computadores, *smart-phones*, *tablets*), o número e tipo de serviços pode aumentar significativamente. O ENUM permite associar a cada número telefônico E.164 uma série de informações adicionais, como, por exemplo, endereços de páginas Web, endereços de e-mail, dentre outros, conforme ilustra a Figura 2.1. Dessa maneira, ao utilizar-se um dispositivo conectado à Internet torna-se possível acessar quaisquer destes serviços, ou outros novos que venham a ser criados.

Neste contexto, o objetivo deste capítulo é apresentar o protocolo ENUM, seu funcionamento e importância no contexto da integração das redes tradicionais e a Internet. O estudo será iniciado por uma breve introdução aos principais conceitos e fundamentos relacionados ao protocolo ENUM e, posteriormente, segue a apresentação deste protocolo.

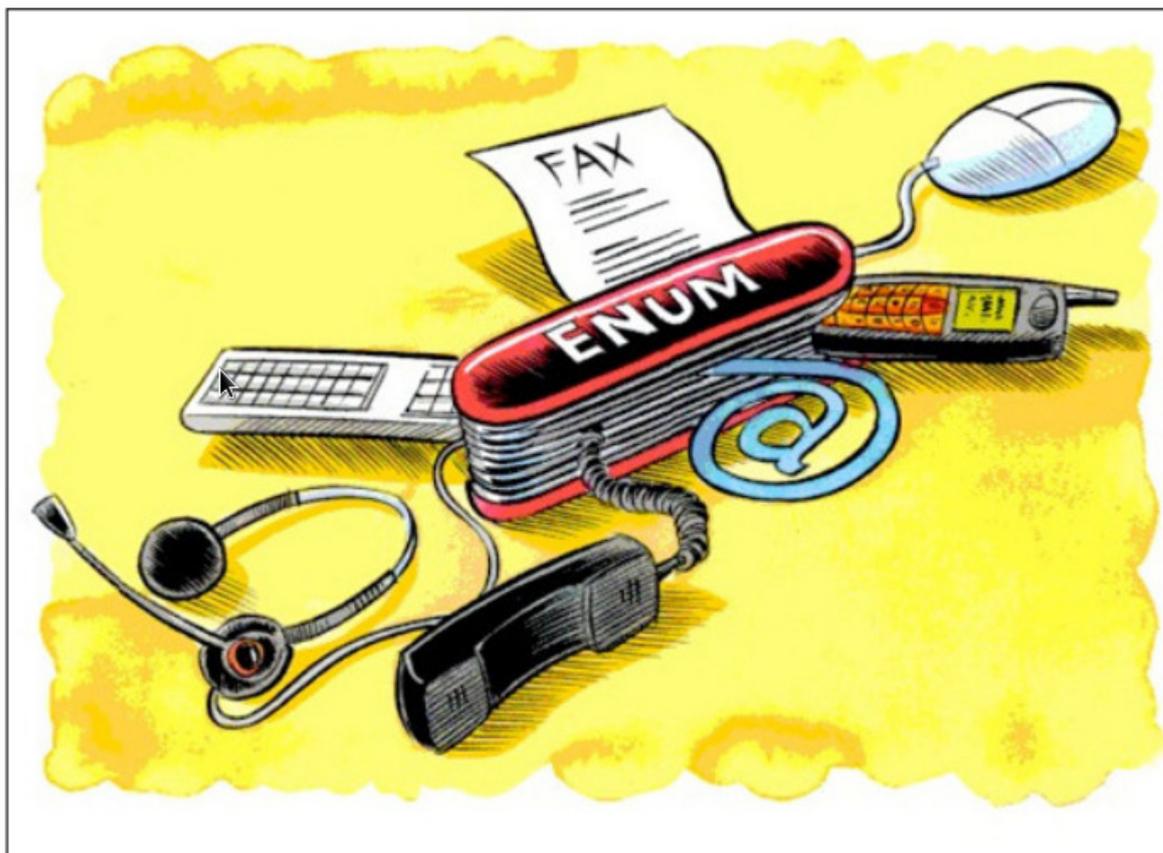


Figura 2.1: O ENUM possibilita a conexão de vários serviços com apenas um único número telefônico E.164 [11].

2.2 *Domain Name System (DNS)*

Endereços IP são de difícil memorização para o homem. Por outro lado, para os roteadores, seu formato fixo e estruturado facilita o processamento de pacotes. Para conciliar estas preferências o serviço DNS foi idealizado para a resolução de um nome em um endereço IP e vice versa [25].

Criado por Paul Mockapetris e inicialmente descrito nos RFCs (*Request For Comments*) 1034 e 1035, o DNS é, simultaneamente, um banco de dados distribuído e implementado em uma hierarquia de servidores de nomes e um protocolo de aplicação que viabiliza consultas a este banco de dados [25]. O DNS existe desde os primórdios da Internet e adquiriu bastante maturidade ao longo dos anos. Logo, a comunidade de pesquisadores escolheu o DNS como o banco de dados para a resolução de números telefônicos E.164 em URIs.

O DNS é um tópico com amplo estudo e, portanto, esta seção faz apenas uma breve explanação dos principais conceitos do DNS utilizados pelo ENUM.

2.2.1 Espaço de Nomes do DNS

Como dito anteriormente, o DNS é um banco de dados distribuído, organizado em uma estrutura hierárquica ramificada de forma análoga a uma árvore. Neste sentido, cada ramificação desta “árvore” dá origem a um nó que representa um domínio pertencente ao banco de dados do DNS, sendo que cada um destes domínios possui um nome composto por uma sequência de rótulos separados por pontos que descrevem o caminho do nó em questão até o nó raiz da árvore DNS. Em cada nível desta hierarquia, encontram-se os servidores de nomes responsáveis por hospedar os domínios deste banco de dados.

No nó raiz da árvore DNS, encontra-se o servidor de nomes raiz responsável pelo domínio raiz. Este domínio possui um rótulo nulo como nome e é denotado apenas por um simples ponto (.). Seguindo na hierarquia, imediatamente após o servidor de nomes raiz encontram-se os Servidores de Alto Nível. Os domínios hospedados neste nível podem ser divididos em:

- Domínios de Alto Nível (*Top Level Domains - TLDs*) genéricos (por exemplo *.com*, *.edu*, *.net*);
- Domínios de Alto Nível por país (por exemplo *.br*, *.us*, *.uk*, *.de*);
- Domínios de Alto Nível para casos especiais (por exemplo *.arpa*).

Logo após os TLDs encontram-se os Domínios de Segundo Nível (*Second Level Domains - SLDs*) como *ufu.br*, *google.com*, etc. A Figura 2.2 ilustra a hierarquia DNS descrita acima.

2.2.2 Zonas

Na seção anterior foi definido que um domínio é dado por rótulos separados por pontos como, por exemplo, *ufu.br*. Como este domínio pode tornar-se demasiadamente grande ele pode ser dividido em subdomínios, tais como, *eletrica.ufu.br*, *mecanica.ufu.br*, *ileel.ufu.br*, etc. A administração do domínio *ufu.br* pode ser realizada em um único servidor de nomes ou pode-se ainda criar servidores de nomes independentes para alguns destes subdomínios.

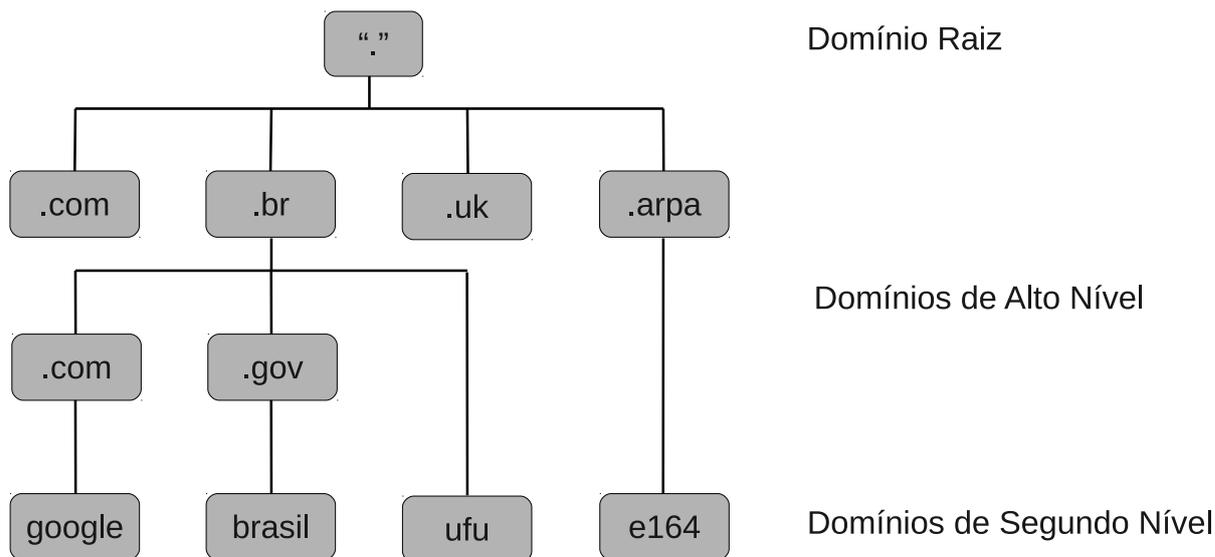


Figura 2.2: Estrutura hierárquica do DNS.

Para cada um destes novos servidores de nomes é estabelecido uma zona DNS. Portanto, uma zona é a parte do espaço de nomes de domínio que é administrada por um servidor de nomes em particular [26].

2.2.3 Resolver

Resolvers são um conjunto de subrotinas utilizadas por uma aplicação cliente (Web, FTP, Telnet) para se comunicar com os servidores de nomes. Os *resolvers* são responsáveis pela realização de consultas a um servidor de nomes, interpretação da resposta e retorno da informação para a aplicação que realizou a requisição.

2.2.4 Registros de Recursos

Registros de Recursos (RR) são registros que definem alguns atributos para um dado nome de domínio, por exemplo, um endereço IP, o servidor de correio eletrônico ou ainda uma *string* com outras informações pertinentes. Os registros de recursos declaram qual é o servidor de nomes responsável por uma dada zona [2].

Dentre os diversos RRs disponíveis é importante destacar o registro NAPTR (*Name Authority Pointer*) que é o tipo de registro utilizado pelo ENUM para a resolução de um número telefônico E.164 em um URI. Detalhes sobre o registro NAPTR são abordados na Seção 2.6.

2.2.5 Servidores de Nomes

O servidor de nomes é um software que é executado em um *host*. Este servidor possui informações para a tradução de nomes de domínio para endereços IP. O servidor de nomes é responsável, portanto, por uma parte do espaço de nomes do DNS, ou seja, uma zona, como elucidado anteriormente.

Existem alguns tipos de servidores de nomes, dentre eles é importante citar:

- Servidor de Nomes Autoritativo: este servidor é responsável por uma ou mais zonas para as quais é considerado autoritativo. As informações sobre esta zona são obtidas a partir da configuração local e não por meio de uma fonte externa;
- Servidor de Nomes *Cache*: este servidor geralmente é um servidor local que obtém informações específicas sobre um domínio na forma de um ou mais registros de recursos, consultando servidores de nomes autoritativos. Estas consultas são realizadas para responder a uma consulta previamente enviada por um *host* cliente ao servidor de nomes *cache*. Sempre que recebe uma resposta a uma requisição realizada, o servidor de nomes *cache* salva esta resposta em sua memória por um período pré-estabelecido pelo servidor autoritativo que enviou a resposta. Este período é conhecido como *Time To Live* (TTL). Ao receber uma nova consulta para este domínio o servidor de nomes *cache* poderá responder a consulta diretamente ao *host* cliente sem a necessidade de consultar o servidor de nomes autoritativo.

2.2.6 Funcionamento do DNS

A Figura 2.3 ilustra um cenário de consulta típico da Internet. Suponha que o *host pollux.ufu.br* deseja encontrar o endereço IP do *host sirius.google.com*. Suponha ainda que o servidor de nomes local para o *host* requisitante seja denominado *dns.ufu.br* e o servidor com autoridade para o segundo *host* seja denominado *dns.google.com*. As consultas ilustradas na Figura 2.3 são descritas a seguir:

- *pollux.ufu.br* envia uma requisição para o servidor de nomes local *dns.ufu.br* com o nome do *host* para o qual deseja o endereço IP;
- Ao receber a requisição o servidor de nomes local faz uma busca em sua memória *cache* para determinar se já possui a resposta requisitada. Supondo que o servidor

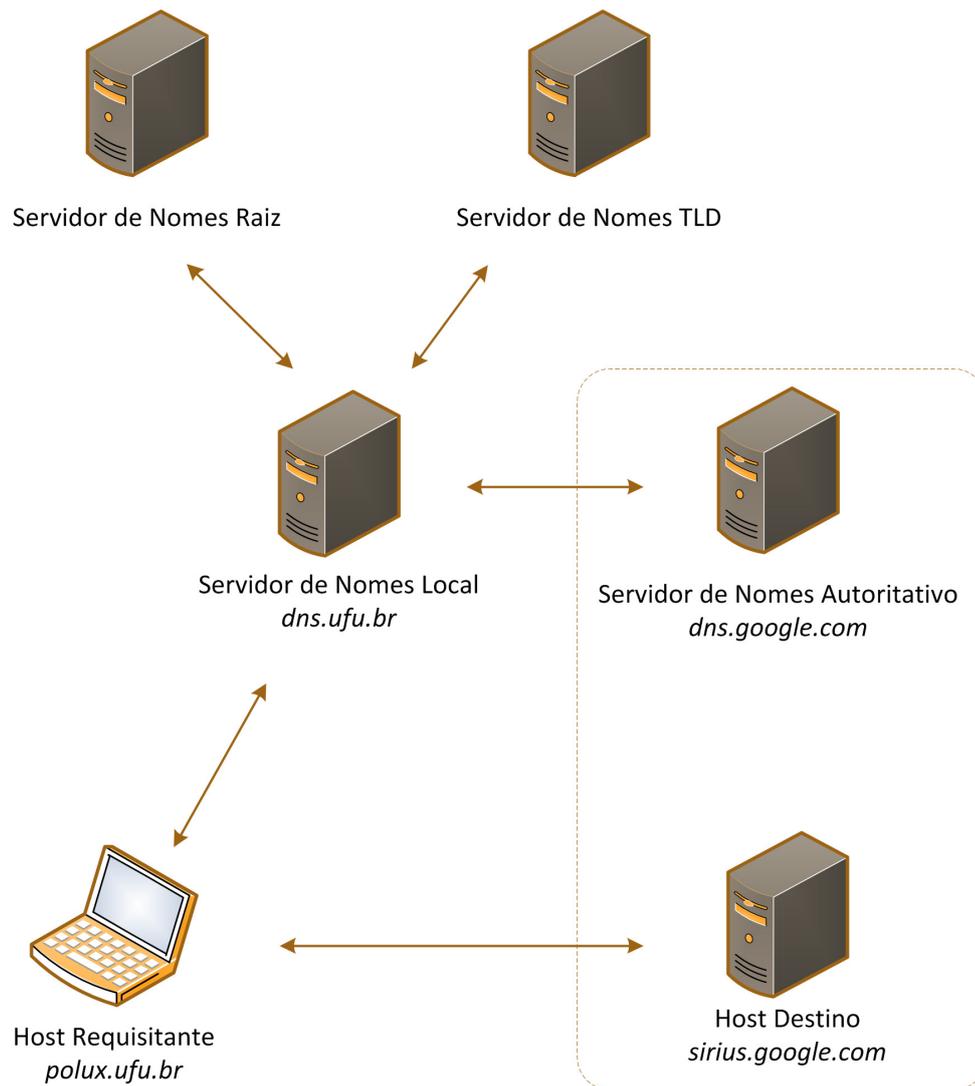


Figura 2.3: Funcionamento típico do DNS.

local não encontrou a resposta desejada, este envia a consulta para o servidor de nomes raiz;

- O servidor de nomes raiz analisa o nome do *host* e identifica, pelo sufixo *.com*, para qual servidor TLD a consulta deve ser repassada;
- O servidor TLD por sua vez reconhece o sufixo *google.com* e responde com o endereço IP do servidor com autoridade para este domínio, ou seja, *dns.google.com*;
- Finalmente, o servidor de nomes local envia a requisição diretamente ao servidor com autoridade para o domínio *google.com* que por sua vez responde com o endereço IP do *host* *sirius.google.com*;

- Ao receber a resposta do servidor autoritativo, o servidor de nomes local a armazena em sua memória *cache* e envia, então, a resposta ao *host* requisitante que por sua vez pode conectar-se com o *host* destino.

Na Figura 2.3 percebe-se que foram necessárias oito etapas para a resolução do nome em um endereço IP. Contudo, se a próxima consulta para este domínio for realizada antes do TTL expirar, a resposta poderá ser obtida com apenas uma iteração, devido ao mecanismo de *cache*.

2.3 Números telefônicos E.164

Números telefônicos E.164 são uma recomendação da *International Communication Union - Telecommunication Standardization Sector* (ITU-T) para o plano de numeração pública internacional. Existem cinco diferentes estruturas para um número telefônico E.164:

- Números telefônicos E.164 para áreas geográficas;
- Números telefônicos E.164 para serviços globais;
- Números telefônicos E.164 para redes de comunicação;
- Números telefônicos E.164 para grupos de países;
- Números telefônicos E.164 para testes (*trials*).

Este trabalho utiliza números telefônicos E.164 para áreas geográficas, uma vez que é o tipo mais comum de número telefônico E.164. Os demais tipos de números telefônicos E.164 não serão explorados. Maiores detalhes sobre a recomendação E.164 da ITU-T podem ser encontrados em [27].

A Figura 2.4 ilustra a estrutura básica de um número telefônico E.164 para áreas geográficas. Um número telefônico E.164 tem um comprimento máximo de 15 dígitos, os quais são distribuídos de acordo com o código do país e o plano nacional de numeração adotado. Para o Brasil, o código do país é igual a 55 e o Código Nacional de Destino (CND) é determinado pelo código de Discagem Direta à Distância (DDD).

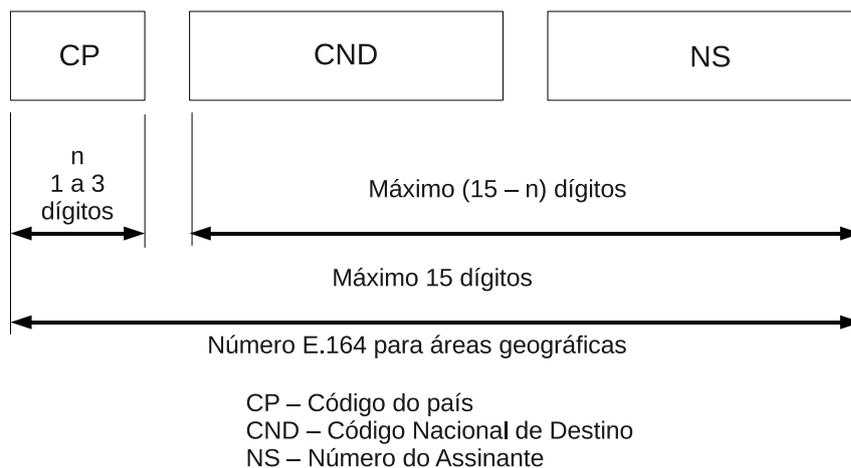


Figura 2.4: Estrutura dos números telefônicos E.164 para áreas geográficas [27].

2.4 Uniform Resource Identifiers (URIs)

Um URI é uma *string* de caracteres compacta utilizada na identificação de um recurso físico ou abstrato na Internet [28]. Sua sintaxe básica é:

Scheme: scheme-specific-part

Alguns exemplos de URIs são:

`http://www.ufu.br`

`sip:saci@perere.com`

`mailto:monster@lochness.com`

Na Internet, todos estes URIs devem ser mapeados em um endereço IP. Esse trabalho é realizado pelo DNS.

2.5 Application Unique String (AUS)

Uma AUS é um número telefônico E.164 menos qualquer caractere que não seja um dígito, exceto pelo caractere '+' que aparece no início do número. O caractere '+' é mantido para evitar ambiguidade com números de telefone que não pertencem ao espaço de nomes E.164 [5]. Para exemplificar o que é uma AUS, considere o número E.164 +(55)-34-3239-1012. A AUS correspondente para este número é dada por +553432391012.

2.6 Registro *Name Authority Pointer* (NAPTR)

Registros NAPTR são utilizados para mapear URIs ou outros registros de recursos (por exemplo o registro SRV). Para o caso do ENUM, este registro é utilizado para a tradução de um número telefônico E.164 em um URI. A seguir encontram-se alguns exemplos de registros NAPTR.

```
NAPTR (
    100                ; order
    10                 ; preference
    "u"                ; flag
    "E2U+sip"          ; service
    "!^(+553432391012)$!sip:user@ufu.br!" ; regexp
    .                  ; no replacement
)
```

```
NAPTR 100 12 "u" "E2U+email:mailto" "!^.*$!mailto:info@ufu.br!" .
```

O campo ('100') estabelece a ordem em que os serviços devem ser processados. O campo preferência ('10') define a prioridade do serviço. Para estes dois campos, quanto menor o valor do campo maior é a ordem/prioridade do serviço. A *flag* 'u' mostra que a resolução do URI indicado no registro NAPTR em um servidor DNS, fornecerá um servidor ou uma lista de servidores que poderão ser contatados diretamente. O próximo campo indica quais protocolos devem ser utilizados para cada serviço, por exemplo, "E2U+sip" indica a utilização do protocolo SIP sobre o protocolo UDP (*User Datagram Protocol*). Os dois últimos campos são *regexp* e *replacement*, respectivamente. Juntos, estes campos formam a Expressão de Substituição do algoritmo DDDS (*Dynamic Delegation Discovery System*). Estes campos são mutualmente exclusivos e, portanto, se um registro retorna valores para ambos os campos considera-se que ocorreu um erro [29]. O campo *regexp* indica ainda que tipo de AUS cada serviço suporta. No exemplo citado acima, o primeiro registro suporta apenas consultas para o número de telefone +553432391012, enquanto o último registro pode ser usado para qualquer tipo de AUS [5].

2.7 Session Initiation Protocol (SIP)

O SIP é um protocolo de sinalização desenvolvido para configurar, modificar e finalizar sessões multimídias. O protocolo SIP pode ser utilizado ainda para requisição e entrega de informações de presença (*status online/off-line* e localização) bem como para sessões de mensagens instantâneas [30].

2.8 Descrição Geral do Protocolo ENUM

2.8.1 O protocolo ENUM no contexto PSTN-VoIP

O número de usuários da tecnologia VoIP tem aumentado significativamente a cada ano, tornando esta tecnologia uma realidade de negócios que cresce rapidamente. Apesar da disponibilização de serviços VoIP por diversos provedores, geograficamente, os usuários desta tecnologia situam-se em um conjunto de “ilhas isoladas”, onde a interconexão entre estas ilhas se realiza por meio da PSTN. A Figura 2.5 ilustra este cenário de troca de tráfego entre as operadoras VoIP por meio da PSTN.

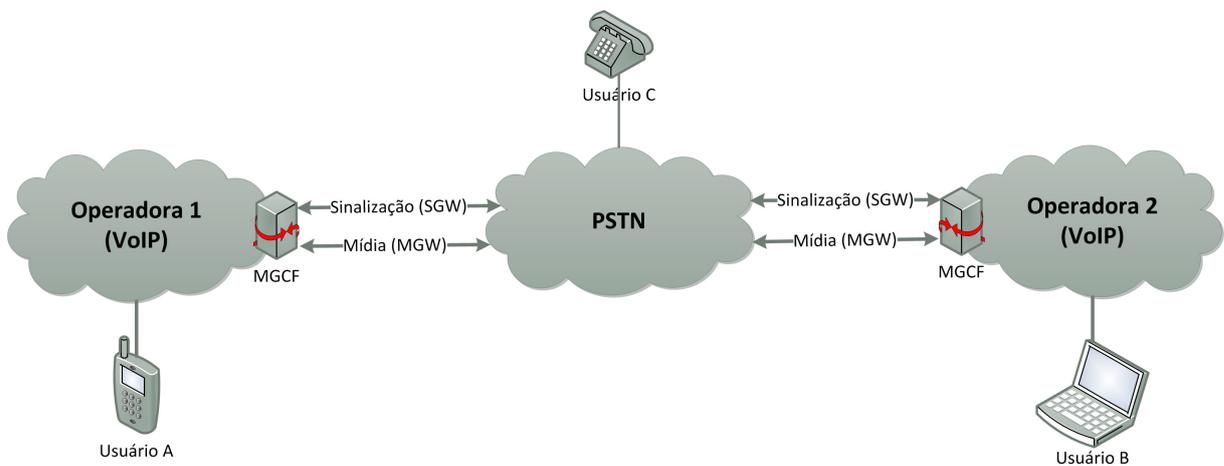


Figura 2.5: Interconexão de Operadoras VoIP por meio da PSTN.

Como exemplo, a Figura 2.5 mostra que, no cenário atual, para que o usuário A possa se comunicar com o usuário B, a Operadora VoIP 1 deve necessariamente estabelecer a chamada por meio da PSTN para alcançar a Operadora VoIP 2. Esta operação é realizada por meio de um *Media Gateway Control Function* (MGCF) e enlaces do tipo E1 que geralmente são utilizados para interconectar as operadoras VoIP à PSTN. Ao receber

uma sessão SIP, o MGCF realiza uma conversão entre o protocolo SIP e os protocolos de sinalização da PSTN utilizando o Gateway de Sinalização (*Signalling Gateway* - SGW) e envia a requisição já convertida para a PSTN. O MGCF também controla o Gateway de Mídia (*Media Gateway* - MGW), que é responsável pela conversão de mídia entre a rede VoIP e a PSTN. O SGW e o MGW atuam tanto no envio de sinalização e mídia para a PSTN como também realizam a conversão inversa para os padrões da rede IP.

Esta estrutura de interconexão demanda investimentos por parte das operadoras VoIP na compra de equipamentos SGW e MGW, no pagamento da Tarifa de Uso de Comutação (TU-COM) e no pagamento de aluguel para utilização dos enlaces do tipo E1. Estes gastos aumentam os custos totais de operação e, conseqüentemente, oneram o usuário final.

Embora sejam reconhecidas as vantagens oferecidas pelas redes VoIP em relação à PSTN, percebe-se que a substituição instantânea de uma tecnologia pela outra não é possível devido à complexidade e custos envolvidos em tal ação. Neste sentido, existirá um período de transição em que as duas tecnologias irão conviver, uma vez que as operadoras de telecomunicações realizaram grandes investimentos na PSTN, que podem ainda não ter sido completamente amortizados, e a migração dos assinantes da PSTN para o serviço VoIP pode ocorrer de forma lenta e gradual.

Neste cenário existem quatro tipos básicos de chamadas telefônicas: PSTN-PSTN, PSTN-IP, IP-PSTN e IP-IP [1]. Logo, o problema não se restringe apenas a evitar o uso da PSTN em chamadas do tipo IP-IP, mas também permitir chamadas entre terminais das duas diferentes redes (PSTN-IP, IP-PSTN), necessárias neste período de transição.

Tradicionalmente, números telefônicos E.164 e URIs são utilizados na PSTN e na Internet, respectivamente, para a identificação de usuários. Protocolos de sinalização como o SIP e o H.323 podem trabalhar com estes dois tipos de identificadores, além de endereços IP. Dessa maneira, para usuários na Internet, que geralmente possuem um teclado alfanumérico disponível, nomes do tipo textual (URIs) são mais utilizados devido à facilidade de memorização e dedução. Contudo, um problema surge quando as redes VoIP e PSTN são interconectadas, uma vez que os terminais da PSTN, tipicamente, oferecem aos usuários um teclado numérico. Sendo assim, é necessário que exista um sistema de mapeamento entre números telefônicos E.164 e URIs que possibilite a interconexão das redes VoIP e PSTN além de permitir chamadas do tipo IP-IP sem a necessidade de utilização da PSTN. Portanto, o modelo de interconexão com DNS/ENUM, mostrado na

Figura 2.6, é uma das soluções mais promissoras para estes problemas.

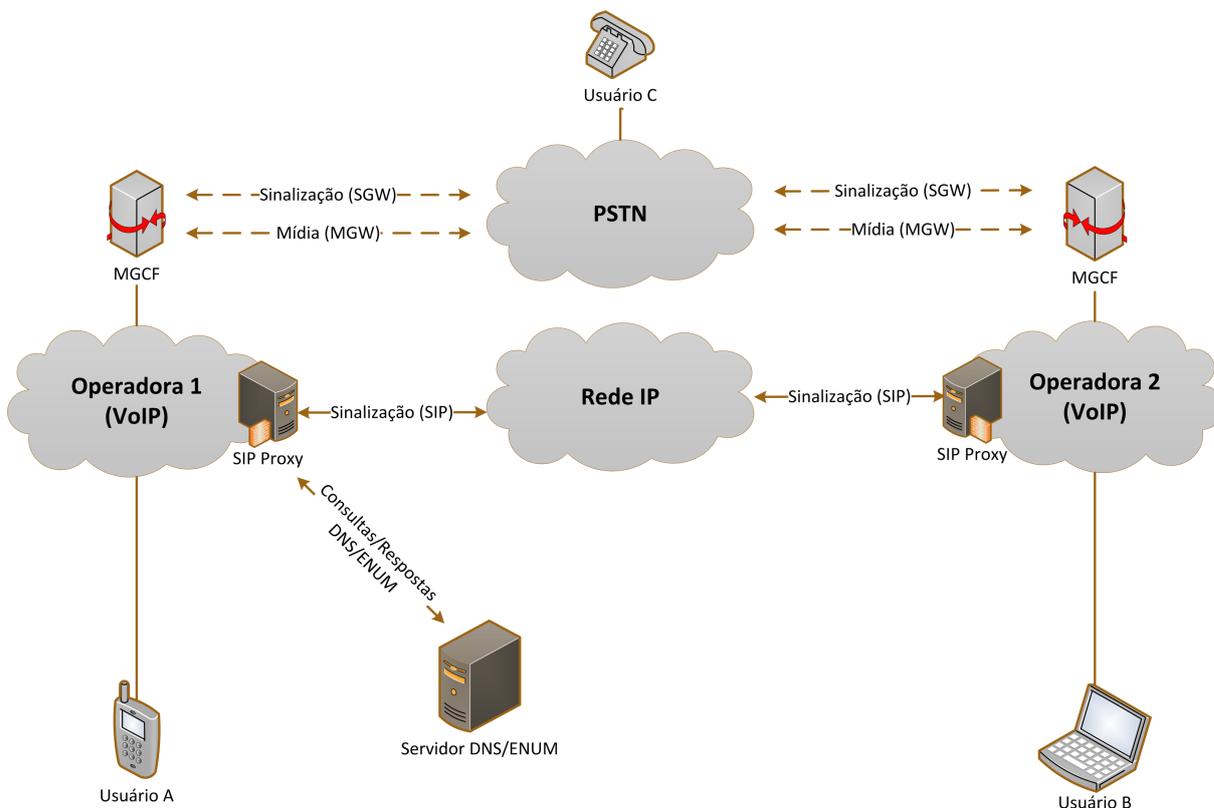


Figura 2.6: Interconexão de Operadoras VoIP por meio de uma rede IP.

O cenário simples ilustrado na Figura 2.6 permite exemplificar o estabelecimento de uma chamada entre os assinantes A e B. Neste cenário, a Operadora VoIP 1 consulta a base de dados DNS/ENUM a fim de encontrar o URI do usuário B (por exemplo, o endereço SIP) correspondente ao número telefônico discado pelo usuário A. Após receber esta informação, a Operadora VoIP 1 utiliza o protocolo SIP para estabelecer uma sessão de comunicação com o usuário B pertencente a Operadora VoIP 2 por meio de um servidor SIP *Proxy*. Dessa forma, a sinalização e a própria chamada poderão trafegar apenas pela rede IP. Os equipamentos MGCF e os enlaces do tipo E1 ainda serão necessários caso o usuário A queira estabelecer uma conexão com o usuário C. A PSTN poderá ainda ser utilizada para conectar as operadoras VoIP em cenários onde a rede IP não ofereça condições mínimas de Qualidade de Serviço.

2.8.2 Aplicações ENUM e o DDDS

O DDDS pode ser definido como um sistema utilizado para implementar uma associação entre *strings* e dados com o objetivo de dar suporte a sistemas de delegação configurados dinamicamente. As funções DDDS mapeiam uma *string* única em dados armazenados dentro de um banco de dados DDDS [31]. Aplicações ENUM utilizam o algoritmo DDDS e o DNS como banco de dados.

A primeira etapa do algoritmo DDDS é conhecida como “*the first well known rule*”. Nesta etapa uma AUS é convertida em uma chave. Esta chave é utilizada como um índice para consultar um banco de dados, que para o caso do ENUM é o próprio DNS e, portanto, essa chave é um FQDN (*Fully Qualified Domain Name*). Para converter a AUS em uma chave única deste banco de dados, a *string* é convertida em um domínio de acordo com o algoritmo a seguir:

- Remove-se todos os caracteres, com exceção dos dígitos numéricos. Por exemplo, para a AUS ‘+553432391012’ a saída é dada por ‘553432391012’. Na prática este passo apenas remove o caractere ‘+’;
- Inverte-se a ordem dos dígitos. Exemplo: ‘210193234355’;
- Insere-se pontos entre cada dígito. Exemplo: ‘2.1.0.1.9.3.2.3.4.3.5.5’;
- Anexa-se a *string* ‘e164.arpa’ ao final. Exemplo: ‘2.1.0.1.9.3.2.3.4.3.5.5.e164.arpa’.

Este FQDN é utilizado para requisição de registros NAPTR. Cada um destes registros contém um destino final (um URI válido apontando para um serviço como SIP ou e-mail), ou ainda uma nova chave em formato de domínio para uma nova consulta ao DNS.

2.8.3 Arquitetura ENUM

Por funcionar sobre a estrutura DNS, o ENUM da mesma forma é baseado em um sistema de delegação estratificado em níveis. É possível destacar duas diferentes abordagens para a arquitetura do ENUM [4]:

- Arquitetura Oficial: Arquitetura definida no RFC 6116 [5] que estabelece a arquitetura a ser usada em escala global pelos usuários. Esta é a arquitetura apoiada pela ITU-T e em uso por diversos países;

- Arquitetura Privada: Arquitetura definida por parte das entidades utilizadoras do ENUM, sejam usuários finais, ou operadoras de telecomunicações.

A arquitetura oficial definida pela ITU-T determina que os diferentes níveis de delegação para o ENUM devem ser definidos por cada política nacional de forma independente. É possível identificar três níveis distintos, descritos abaixo:

- *Tier-0*: administração e gerenciamento do domínio ENUM (*e164.arpa*). Este nível é gerenciado por uma entidade de registro internacional que possui ponteiros para as entidades de registro nacionais no *Tier-1* [32];
- *Tier-1*: administração e gerenciamento do ENUM em cada país. As entidades de registro nacionais neste nível possuem ponteiros para os servidores autoritativos onde os registros NAPTR estão armazenados;
- *Tier-2*: provê o serviço ENUM. Este nível é gerenciado por uma operadora e possui os registros NAPTR. Ele pode ser dividido em dois: no primeiro subnível (*Tier-2A*) os números dos assinantes são divididos de acordo com a operadora que os controla. No segundo subnível (*Tier-2B*) têm-se, enfim, os registros NAPTR.

Os níveis zero e um são naturalmente monopólios das entidades reguladoras como o *Réseaux IP Européens Network Coordination Centre* (RIPE-NCC) e o Comitê Gestor da Internet no Brasil (CGI.br). O nível dois está aberto à concorrência entre as empresas de telecomunicações.

2.8.4 Tipos de ENUM

Como mencionado anteriormente, o ENUM utiliza o DNS como banco de dados. Como em todo banco de dados, é necessário estabelecer os direitos de acesso/alteração das informações. No DNS clássico o banco de dados é público, ou seja, qualquer pessoa pode obter informações armazenadas no banco de dados. Contudo, apenas proprietários de nomes de domínios estão habilitados a realizar alterações nas informações. Para o caso do ENUM, tais questões não estão totalmente esclarecidas [15]. Em função destas indefinições sobre os direitos de acesso/alteração dos dados e das distintas arquiteturas que podem ser utilizadas, surgiram diferentes tipos de ENUM, descritos a seguir [4]:

- *User* ENUM: é uma base de dados pública global que permite que as informações possam ser acessadas e modificadas pelo usuário de forma autônoma utilizando a arquitetura oficial da ITU-T. A utilização do *User* ENUM apresenta diversas dúvidas (principalmente por parte das operadoras) acerca da sua escalabilidade, segurança, privacidade e qualidade dos dados;
- *Infrastructure* ENUM: refere-se à utilização do ENUM pelas operadoras de telecomunicações. Somente a operadora está habilitada a realizar alterações no banco de dados, mas por outro lado, o acesso às informações é livre para qualquer usuário da rede IP pública. Este tipo de ENUM não tem sido bem recebido por parte das operadoras, pois expõe informação crítica da sua rede, pondo em risco a segurança e a operacionalidade da infraestrutura de telecomunicações. Adicionalmente, não existe consenso sobre qual arquitetura utilizar para suportar este serviço;
- *Carrier* ENUM: apresentando-se como o tipo mais promissor, o *Carrier* ENUM destina-se à utilização do ENUM pelas operadoras de telecomunicações para operações de suporte à rede num ambiente protegido utilizando uma arquitetura ENUM privada. Neste tipo de ENUM, as operadoras ou grupos de prestadores de serviço compartilham informações de clientes para o suporte à suas relações de *peering*;
- *Private* ENUM: refere-se a utilização do ENUM pelos usuários finais utilizando uma arquitetura ENUM privada. Os usuários têm livre acesso às informações do banco de dados privado. Este tipo de ENUM têm sido proposto como uma alternativa ao *User* ENUM de forma a ultrapassar as limitações impostas por agências reguladoras.

2.8.5 Funcionamento do ENUM

A Figura 2.7 ilustra um exemplo de resolução de um número telefônico E.164 quando os interlocutores pertencem ao mesmo domínio. Suponha que o usuário A deseja comunicar-se com o usuário B. Quando A disca o número de B, a rede telefônica encaminha esta requisição de chamada para um *gateway*. O *Gateway* SIP, então transforma o número telefônico E.164 em um FQDN, e envia uma consulta DNS NAPTR para o servidor ENUM local. O servidor ENUM local retorna os dois registros NAPTR, exemplificados a seguir.

```
IN NAPTR 100 50 "u" "E2U+sip" "!^(\+553432561010)$!sip:\\user-b@ufu.br!" .
IN NAPTR 100 51 "u" "E2U+h323" "!^(\+553432561010)$!h323:user-b@ufu.br!" .
```

Segundo as ordens e preferências, a operação de expressão regular é aplicada e um URI do tipo *sip:user-b@ufu.br* é gerado. Ao receber o URI, o *Gateway SIP* envia uma consulta ao servidor DNS local para traduzir o domínio contido no URI (*ufu.br*) em um endereço IP. O resultado da consulta é o endereço IPv4 do servidor *SIP Proxy* para aquele domínio. O *Gateway SIP* abre, então, uma sessão UDP na porta '5060' deste servidor *SIP Proxy* para completar a chamada (*call setup*), e requisita uma sessão de voz com o usuário B deste servidor.

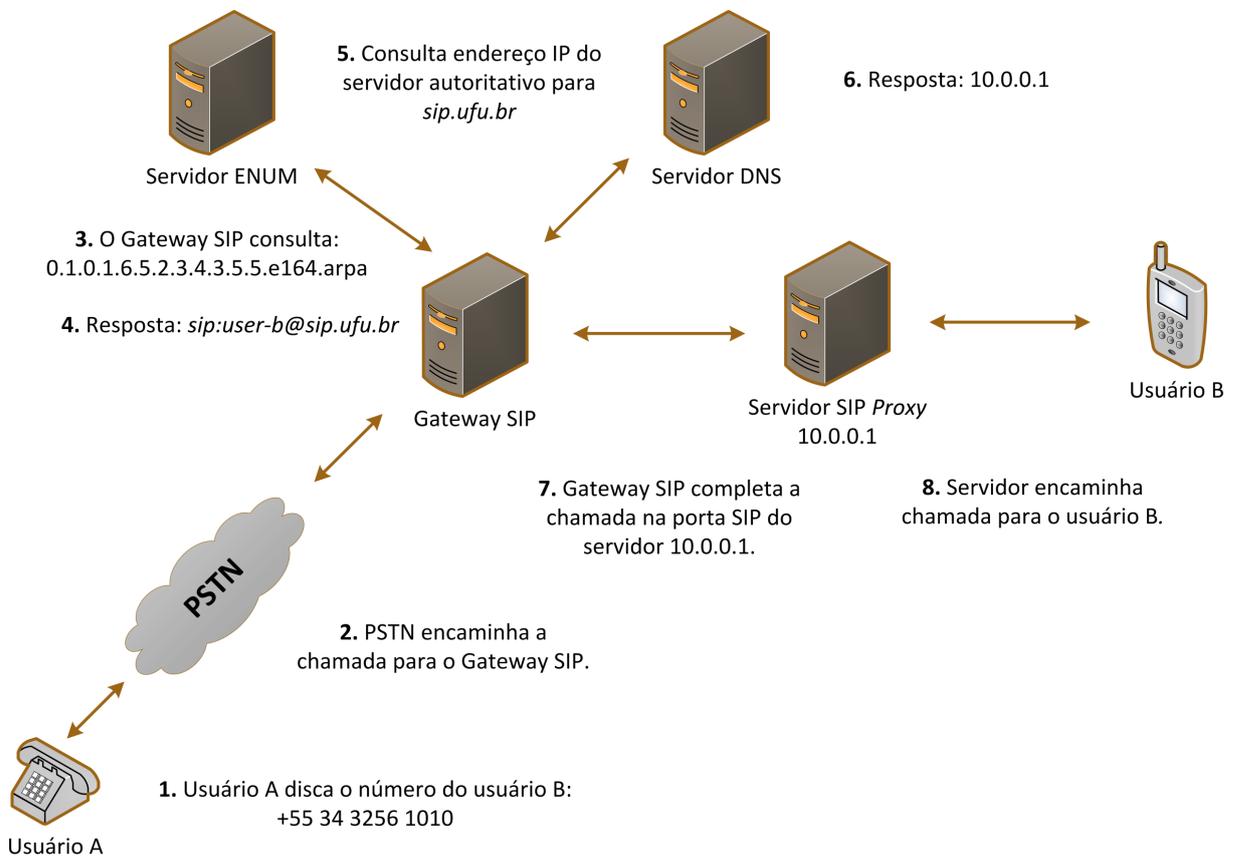


Figura 2.7: ENUM: exemplo de resolução de um número telefônico E.164.

2.9 Considerações Finais

Os conceitos apresentados neste capítulo são importantes para o entendimento do protocolo ENUM bem como do próprio simulador proposto nesta dissertação. Embora

o ENUM seja baseado na estrutura DNS existente, ele possui algumas características próprias como:

- gigantesco espaço de nomes;
- exigência de pequenos tempos de resolução dos números telefônicos E.164 para permitir desempenho semelhante ao da PSTN;
- maior tráfego na rede IP, uma vez que o tamanho da mensagem de resposta às consultas DNS/ENUM é maior que no DNS Clássico.

Estas características impõem aos atuais servidores de nomes uma nova realidade quanto aos requisitos de desempenho, o que demanda análises sobre o comportamento dos servidores de nomes diante destas novas exigências.

O capítulo seguinte é dedicado a descrição do simulador proposto, que tem por objetivo fomentar e facilitar os estudos sobre o comportamento dos servidores de nomes.

Capítulo 3

Uma Proposta de Plataforma de Simulação para o Estudo do Protocolo ENUM

3.1 Introdução

Como foi dito nos capítulos anteriores, o protocolo ENUM vem se destacando como importante solução para os problemas relacionados ao uso da PSTN para interligar operadoras VoIP, bem como oferecer a possibilidade de convivência destas duas tecnologias durante o período de transição. Este capítulo apresenta uma proposta de plataforma de simulação que visa oferecer um ambiente para o estudo do protocolo ENUM baseando-se no comportamento dos servidores de nomes autoritativos atuando em um contexto ENUM, isto é, sob tráfego de consultas para a tradução de um número telefônico E.164 em um URI.

3.2 Descrição do Problema

Na Seção 1.1 destacou-se que apesar do ENUM ser uma tecnologia relativamente madura, integradora e que, potencialmente, reduzirá os custos de operação das redes de telecomunicações, a implantação deste protocolo acontece gradualmente e de acordo com o interesse de cada país nessa tecnologia. Ainda existem muitos desafios para o ENUM se consolidar como protocolo de interconexão entre as redes de comunicações existentes,

devido tanto a questões de regulamentação e modelo de negócios para as operadoras, quanto a quesitos técnicos relativos ao funcionamento e desempenho do protocolo.

Ao utilizar a estrutura do DNS para mapear números telefônicos E.164 em URIs, o protocolo ENUM insere novos desafios a este sistema devido a alguns requisitos mais restritivos que os presentes no DNS clássico. Citando brevemente alguns destes desafios, tem-se:

- Estabelecer as chamadas em um intervalo similar ao oferecido pela PSTN, levando em consideração que o tamanho da resposta com registros NAPTR é geralmente maior que outros registros DNS típicos;
- Oferecer atualização de registros de forma rápida e sem a interrupção do serviço;
- Disponibilizar dispositivos de segurança que assegurem o sigilo dos dados armazenados no banco de dados.

Diante destes desafios, é importante conhecer o comportamento do protocolo e dos servidores diante de diferentes cenários de operação. A realização de medições em um sistema real para efetuar tais avaliações, é a primeira opção a ser considerada. Entretanto, nem sempre é possível aplicar tal metodologia, devido à falta de equipamentos, custos envolvidos e, até mesmo, em função da não existência de uma rede implementada, como, por exemplo, em estudos para implantação inicial do serviço. Neste contexto, o estudo dos parâmetros por meio de modelagem e simulação surge como uma solução viável e confiável. Ressalta-se que em um ambiente de simulação é possível construir diferentes cenários e controlar os parâmetros de acordo com o objetivo do estudo e encontrar resultados próximos aos de um sistema real por um custo bem menor.

Ressalta-se também que, como descrito na Seção 1.2, existem poucos trabalhos desenvolvidos nesta área, gerando uma carência por ferramentas de simulação que facilitem o estudo do protocolo ENUM.

Diante do que foi exposto acima, esta dissertação apresenta uma proposta para solucionar o problema do estudo dos principais parâmetros de influência em uma implementação do protocolo ENUM. A solução proposta é baseada no desenvolvimento de uma plataforma de simulação que possibilite ao usuário a construção de um ambiente onde é possível controlar e avaliar tais parâmetros.

3.3 Descrição da Plataforma de Simulação

3.3.1 Considerações Iniciais e Simulador Protótipo

O projeto de concepção da plataforma de simulação iniciou-se com uma varredura nos trabalhos relacionados ao protocolo ENUM. Em [24], foi possível encontrar o desenvolvimento de uma plataforma de simulação baseada na arquitetura ENUM implantada na França. Infelizmente, não foi possível ter acesso a este simulador, mas apenas a parte do código fonte, além da descrição do desenvolvimento da plataforma, apresentada em [2].

Diante dessa realidade, inicialmente, foi criado um projeto piloto de um simulador, desenvolvido para ser integrado ao NS-2 e que utilizava os dados de medições realizadas no sistema ENUM francês, que foram registrados em [2]. Os resultados de simulação obtidos a partir deste simulador piloto estão indicados em [33]. A Figura 3.1 ilustra a topologia utilizada no simulador.

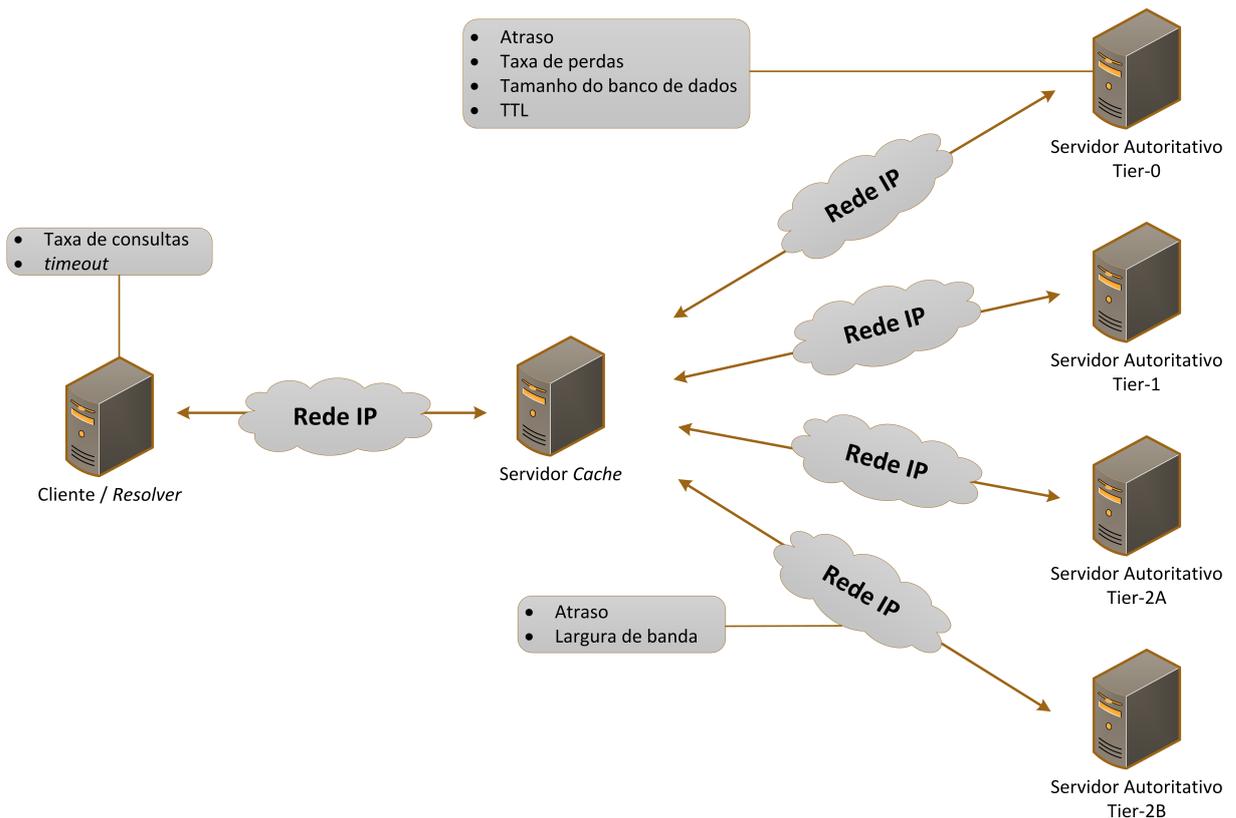


Figura 3.1: Topologia de testes utilizada no protótipo da plataforma de simulação [33].

O módulo cliente é um gerador de tráfego responsável por enviar requisições para o módulo *resolver*. O intervalo entre consultas sucessivas é exponencialmente distribuído e

depende da taxa de consultas informada pelo usuário.

O módulo *resolver* é encarregado de enviar consultas recursivas ao servidor de nomes *cache*. Este módulo também é responsável por obter os dados estatísticos da simulação, tal como o tempo global de resposta estimado, a taxa de perdas e o número de saltos necessários para a tradução do número telefônico E.164 em um URI, etc.

O módulo servidor *cache* recebe uma consulta recursiva enviada pelo módulo *resolver* e procura a resposta em sua memória cache. Caso o servidor possua informações suficientes para resolver a consulta (*cache hit*), ele responde imediatamente para o *resolver*. Em caso contrário, se a informação não está disponível (*cache miss*), o servidor repassa a consulta para o servidor de nomes autoritativo adequado. Após receber o registro NAPTR, o servidor *cache* envia a resposta ao *resolver* e armazena este registro em sua memória *cache* de acordo com o campo TTL ajustado pelo servidor de nomes autoritativo.

O módulo servidor autoritativo é distribuído em quatro níveis de acordo com a arquitetura adotada. Este módulo responde às requisições enviadas pelo servidor *cache* com o endereço do próximo nó que tem informações sobre a consulta em questão. Isso ocorre até que o servidor autoritativo para tal domínio que possui o registro NAPTR seja encontrado. Estes servidores foram baseados no software BIND com os mesmos arquivos de configuração e parâmetros de desempenho. O atraso de processamento e a taxa de perdas para cada um dos servidores autoritativos foram definidas de acordo com a metodologia apresentada em [22], além dos resultados das medições realizadas no sistema ENUM francês, apresentados em [2].

Os resultados da simulação mostraram que este simulador já apresentava melhorias em relação ao simulador proposto em [24] como, por exemplo, na faixa de análise dos parâmetros de estudo. A Tabela 3.1 mostra uma comparação entre os limites de alguns parâmetros do simulador apresentado em [24] e o simulador desenvolvido no projeto piloto, apresentado em [33].

Tabela 3.1: Comparação do desempenho dos simuladores proposto em [24] e [33].

Simulador	[24]	[33]
consultas por segundo (qps)	1 - 254	1 - 50000
banco de dados	1 - 36000	1 - 1000000

Ao observar os dados apresentados na Tabela 3.1, é possível identificar que o simulador apresentado em [33] ampliou significativamente a faixa de valores dos parâmetros de estudo. Dessa forma, este simulador forneceu novos resultados de simulação, não contemplados em [24], possibilitando o estudo do desempenho dos servidores diante de alguns cenários, nomeadamente:

- Desempenho considerando diferentes valores para o TTL;
- Desempenho considerando o número de saltos para a resolução do número E.164;
- Desempenho considerando diferentes valores para o *timeout* no *resolver*;
- Desempenho considerando diferentes valores para a largura de banda dos enlaces.

Embora apresentasse uma evolução em relação ao simulador apresentado em [24], o simulador protótipo apresentava também algumas restrições, tais como:

- O simulador foi desenvolvido para ser integrado ao NS-2, que por sua vez está recebendo poucas atualizações e manutenção. Além disso, o NS-2 não utiliza técnicas modernas de programação, como por exemplo os *smart pointers*, fazendo com que o consumo de memória seja alto, precarizando o desempenho geral do simulador;
- O simulador baseia-se nas medições realizadas no sistema ENUM francês. Atualmente, não existem indícios da continuação deste projeto francês. Além do mais, o hardware e software utilizados nos testes apresentados em [2] estão desatualizados. Existe ainda a possibilidade de erros nas medições, ou de interpretação errônea destes dados, uma vez que estes não foram apresentados de forma clara.

Diante dos fatores explicitados acima, e considerando que não foi possível ter acesso a todos os dados dos testes realizados no sistema real francês, não foi possível validar o simulador protótipo de forma satisfatória.

Neste contexto, o projeto de concepção da plataforma de simulação foi feito para contornar os problemas apresentados pelo projeto piloto. Dessa forma, este novo projeto contempla os pontos positivos do projeto piloto, além de propor soluções para as deficiências identificadas no simulador protótipo.

Neste sentido, as próximas subseções apresentam as decisões de projeto realizadas para que o simulador fosse desenvolvido com sucesso.

3.3.2 Definição do Sistema Real

De acordo com [34], a simulação pode ser definida como o processo de construção de um modelo de um sistema real e a condução de experimentos com este modelo, com o objetivo de entender o comportamento do sistema, ou ainda avaliar diferentes estratégias de operação para este sistema. Neste sentido, um modelo de simulação é tipicamente utilizado para reproduzir o comportamento de um sistema real. Geralmente, emprega-se técnicas de validação para assegurar que o modelo descreve corretamente o sistema real. Partindo deste pressuposto, o desenvolvimento da plataforma de simulação proposta neste trabalho foi realizado a partir da construção de modelos para cada elemento pertencente ao sistema real de interesse, o qual foi escolhido de forma a facilitar a validação dos modelos propostos, por meio da comparação dos resultados obtidos no sistema real e os advindos do simulador.

Inicialmente, é importante definir o sistema real que está sendo considerado. Neste sentido, a Figura 3.2 ilustra um diagrama simples de um cenário de consultas do protocolo ENUM.

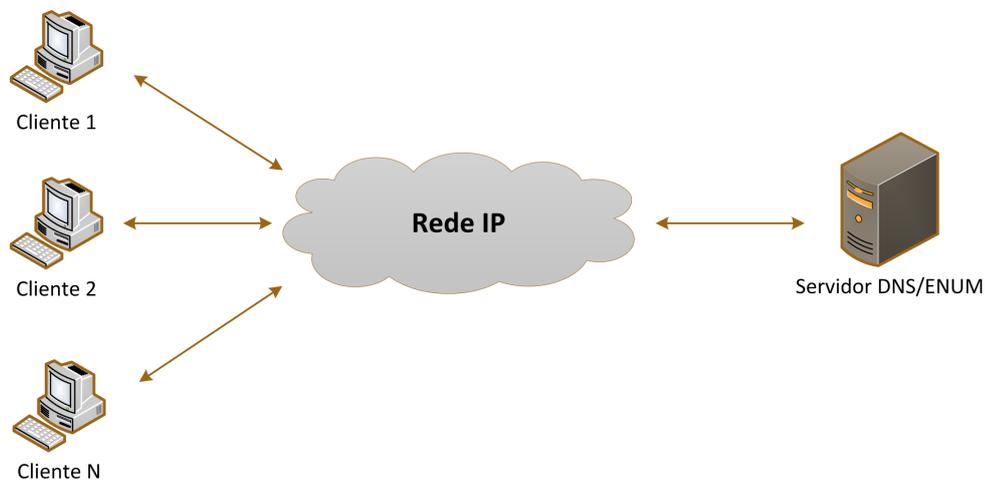


Figura 3.2: Diagrama simples de um cenário de consultas do protocolo ENUM.

Na Figura 3.2, o sistema real é dado por uma série de máquinas clientes que realizam consultas a um servidor DNS/ENUM por meio de uma rede IP. Este é um cenário caro e complexo de ser reproduzido em laboratório e, portanto, resultados de testes realizados neste sistema real não serão facilmente obtidos. É necessário, então, simplificar este sistema a fim de facilitar a obtenção de resultados em um sistema real.

Para realizar esta simplificação, é necessário estabelecer os parâmetros que serão con-

siderados no estudo do protocolo ENUM. Para o presente trabalho, definiu-se que os parâmetros avaliados são a vazão proporcionada pelo servidor DNS/ENUM, o tempo gasto por este servidor para responder a uma consulta e a capacidade do servidor de ser escalável, isto é, o desempenho do servidor diante de diferentes tamanhos de banco de dados. A avaliação destes parâmetros possibilitará conhecer o comportamento do servidor DNS/ENUM e procurar respostas para alguns dos desafios listados nas Seções 1.1 e 3.2, principalmente os relacionados com a vazão, a escalabilidade e o tempo de estabelecimento da chamada. O simulador foi desenvolvido de forma que os demais desafios listados possam ser estudados futuramente com a integração de novas funcionalidades.

Uma vez que, a priori, o interesse é conhecer o comportamento do servidor DNS/ENUM com relação a vazão de respostas às consultas, o tempo de resposta e a escalabilidade, pode-se propor um cenário mais simples que o apresentado na Figura 3.2, o qual é apresentado na Figura 3.3. Para a realização dos testes laboratoriais, utilizou-se uma rede IP controlada e as diversas máquinas clientes foram substituídas por uma única máquina, a qual emula todas as requisições realizadas pelos clientes. Dessa maneira, tem-se um cenário de avaliação de desempenho que possibilita a obtenção de resultados reais a serem utilizados na validação do simulador.



Figura 3.3: Diagrama simples do sistema real avaliado em laboratório.

Portanto, a modelagem utilizada no desenvolvimento da plataforma de simulação proposta é baseada no sistema ilustrado na Figura 3.3.

3.3.3 Ambiente de Implementação do Simulador

Para realizar a implementação da plataforma de simulação, necessita-se escolher um ambiente ou ferramenta de desenvolvimento adequada. A escolha da ferramenta/linguagem de programação a ser utilizada para conceber o modelo conceitual tem um impacto significativo tanto no tempo de desenvolvimento quanto na eficiência do simulador [35].

A elaboração completa de um simulador, a partir de uma linguagem de programação como o C++, Java ou Python, torna-se uma tarefa difícil já que será necessário desenvolver

toda a representação da rede de comunicação em seus mínimos detalhes.

Dessa maneira, ao se utilizar um ambiente de simulação, tal como o NS-2, pode-se aproveitar todo o trabalho já realizado por outros pesquisadores na implementação da ferramenta de simulação e adicionar apenas os novos módulos de simulação.

Dentre os ambientes de simulação disponíveis é possível destacar:

- *Network Simulator 2* (NS-2): O NS-2 é um ambiente de simulação orientado à eventos (*event-driven*), que possibilita simulações de funções e protocolos de redes cabeadas e sem fio (por exemplo, algoritmos de roteamento, TCP, UDP) [34]. O NS-2 é baseado em duas linguagens, a saber, C++ e *Object-oriented Tool Command Language* (OTcl). O código em C++ é responsável por modelar o comportamento dos nós de simulação, enquanto o OTcl é utilizado em *scripts* que controlam a simulação, como na definição da topologia da rede. Apesar de prejudicar o desempenho do simulador, esta escolha foi realizada para evitar recompilações desnecessárias, uma vez que, na época em que o NS-2 foi concebido, em 1996, as máquinas tinham pequeno poder de processamento [36]. Atualmente, esta estratégia já não é mais tão atraente ou necessária;
- *Network Simulator 3* (NS-3) : Como o seu predecessor, o NS-3 é baseado no C++ para a implementação dos modelos de simulação. Entretanto, o NS-3 abandonou o uso do OTcl para os *scripts* de controle. Os *scripts* podem ser escritos em C++ ou em Python. Um dos objetivos fundamentais do NS-3 é o aperfeiçoamento do realismo dos modelos, isto é, a implementação dos modelos de simulação deve ser próxima da implementação real do software/protocolo que estes modelos representam [37]. Além disso, o NS-3 integra alguns conceitos de arquitetura do GTNetS [38], que um é simulador com boas características de escalabilidade. Estas decisões de projeto custaram a compatibilidade do NS-3 com o NS-2 e, portanto, os módulos escritos para o NS-2 devem ser reescritos para serem portados para o NS-3;
- OMNET++: Em contraste com o NS-2 e o NS-3, o OMNET++ não é simulador de rede por definição, mas um ambiente de simulação de propósito geral. As simulações de rede ficam a cargo do *framework* INET, que possui um ampla coleção de modelos de protocolos da Internet. Existem ainda, outros *frameworks* que possibilitam estudos de redes sem fio e móveis. Assim como o NS-2 e o NS-3, o OMNET++ é baseado

no C++ para a implementação de seus modelos. Além disso, existem extensões para simulações em tempo real, emulação de rede, suporte para outras linguagens (Java, C#), comunicação com banco de dados, dentre outras funcionalidades [37].

Embora possua uma ampla coleção de módulos desenvolvidos ao longo dos anos pela comunidade acadêmica, percebe-se que o NS-2 recebe poucas atualizações e manutenção, sendo que todos os esforços neste sentido estão direcionados para o NS-3.

Além do mais, segundo [36], o NS-3 e o OMNET++ são capazes de realizar simulações de extensas redes de forma eficiente e, portanto, se apresentam como boas escolhas de ambiente de simulação.

Neste contexto, o NS-3 foi escolhido como ferramenta de desenvolvimento e análise a ser utilizada nesta dissertação, visando a avaliação de parâmetros de influência na implantação do protocolo ENUM. Tal escolha se justifica pela maior familiarização com este ambiente e o bom suporte oferecido pela comunidade de usuários deste simulador, o que facilita a integração de novos módulos.

3.4 Considerações Finais

Diante do problema descrito na Seção 3.2, este capítulo foi dedicado a uma apresentação geral da plataforma de simulação proposta. Neste sentido, uma das primeiras considerações de projeto foi a definição do sistema real em que o simulador será baseado. Visando a reprodução deste sistema real em laboratório, estabeleceu-se um sistema simples que permite maior controle dos parâmetros de análise.

Outra importante consideração de projeto, foi a determinação do ambiente de implementação do simulador. Diante das opções disponíveis, o NS-3 foi considerado o ambiente mais promissor para tal tarefa, devido ao seu desempenho quanto à escalabilidade e aspectos relativos a manutenção, suporte, documentação e familiarização, o que contribui para a integração de novos módulos com maior praticidade.

O próximo capítulo é dedicado à descrição da metodologia para os experimentos realizados em laboratório, bem como a apresentação dos resultados obtidos nestes testes.

Capítulo 4

Descrição dos Testes em Laboratório

4.1 Introdução

Um modelo, em engenharia, pode ser definido como uma representação de um sistema com o intuito de estudá-lo. Para a maioria dos casos, é necessário somente considerar os aspectos do sistema que afetam esse estudo. Estes aspectos são representados no modelo do sistema, e este modelo, por definição, é uma simplificação do sistema. Por outro lado, o modelo deve ser suficientemente detalhado para permitir conclusões válidas sobre o sistema real [39]. Neste sentido, uma das formas de assegurar que o comportamento do modelo está em consonância com o sistema real se baseia na realização de testes cujo objetivo é a validação da modelagem. Neste contexto, uma opção poderosa e eficaz, é a comparação entre os resultados obtidos no sistema real e aqueles advindos da simulação [40].

Neste contexto, procedeu-se a realização de testes em laboratório a fim de se obter resultados que serão utilizados posteriormente na validação do modelo de simulação. Os detalhes dos testes realizados estão descritos nas seções seguintes.

4.2 Perfil do Hardware das Máquinas

Para a realização dos testes de desempenho foram escolhidas duas máquinas com hardware de alto desempenho, cujo perfil está descrito a seguir:

- Dois processadores Intel® Xeon® 6-Core E5645 HT - 2.40 GHz, 12 MB de *cache*;
- Placa mãe Intel® Server Board S5500BCR (Bluff Creek);

- 16 GB de memória RAM;
- Dois discos rígidos de 500 GB;
- Duas portas de rede Gigabit Ethernet.

A interligação destas máquinas foi realizada com um Switch 3Com 4228G 24x 10/100 Mbps + 2x 10/100/1000 Mbps.

4.3 Softwares

4.3.1 Sistema Operacional

O sistema operacional utilizado na máquina cliente foi a distribuição Linux Ubuntu 11.10. O sistema operacional utilizado na máquina servidora foi o Ubuntu Server 11.10.

4.3.2 Software DNS/ENUM

Existe uma grande variedade de softwares para servidores DNS/ENUM. Em [41], é possível encontrar uma lista com os principais softwares DNS/ENUM disponíveis atualmente no mercado. Dentre estas possibilidades o software DNS/ENUM escolhido para os testes foi o software BIND (*Berkeley Internet Name Domain*). O software BIND [42] é mantido e distribuído pelo *Internet Software Consortium* (ISC) sob a licença BSD. Este software é de fato o software DNS mais utilizado atualmente e em todos os níveis da hierarquia de servidores DNS da Internet. Disponível para diversas plataformas e com ampla documentação, o software BIND pode atuar como servidor de nomes autoritativo e servidor de nomes *cache* e possui todas as funcionalidades especificadas nos RFCs para o DNS. Apesar de utilizar tradicionalmente um sistema de banco de dados baseado em arquivos texto conhecidos como *zone files*, o software BIND oferece em suas últimas versões a possibilidade de utilização de outros tipos de banco de dados (MySQL, ODBC, LDAP, etc) através da extensão DLZ BIND. O software BIND contempla também questões de segurança oferecendo suporte a DNSSEC e TSIG. Todas estas características fizeram com que o software BIND fosse o mais adequado para os testes laboratoriais. A versão do software BIND utilizada foi a 9.8.1, que é a versão estável mais atual até o presente momento.

4.3.3 *EnumBenchTool*

A *EnumBenchTool* é uma ferramenta para gerenciamento de teste de *benchmarking* de softwares DNS/ENUM desenvolvida pelo Laboratório de Redes de Computadores da Universidade Federal de Uberlândia. Esta ferramenta foi desenvolvida na linguagem Python com o objetivo de automatizar, uniformizar e validar os testes, além de facilitar a obtenção dos resultados.

A *EnumBenchTool* está em estágio inicial de desenvolvimento e atualmente oferece suporte aos seguintes softwares: BIND, MyDNS-NG, NSD e PowerDNS. É importante ressaltar que no estágio atual, a *EnumBenchTool* não é a responsável pelo teste de *benchmarking* propriamente dito. Na verdade, a *EnumBenchTool* empacota uma série de outras ferramentas existentes, facilitando o gerenciamento dos testes de *benchmarking* ao tornar o processo de configuração, sincronização e validação transparente para o usuário. A seguir encontra-se uma breve descrição de cada uma destas ferramentas gerenciadas pela *EnumBenchTool*.

4.3.3.1 *dnsperf*

Para realizar os testes de desempenho é necessário emular o tráfego de clientes consultando o servidor DNS/ENUM. O gerador de tráfego escolhido foi a ferramenta *dnsperf* [43]. Esta ferramenta é um software desenvolvido pela empresa Nominum, amplamente utilizado para análise de desempenho de servidores DNS autoritativos. Esta ferramenta pode emular vários clientes simultaneamente em um único processo. Cada cliente trabalha em um sistema em *loop* fechado onde uma consulta só é realizada depois que a resposta para a consulta anterior tenha sido processada. Esta ferramenta oferece uma série de opções de configuração sendo possível ajustar valores para o número de clientes simultâneos em cada processo, o intervalo de tempo do teste, o número máximo de consultas dentre outros [44]. A *EnumBenchTool* é responsável por engatilhar a *dnsperf* a cada etapa de teste, informando qual o número de clientes que deve ser emulado, o tempo de execução desta etapa de testes, o arquivo com as consultas dentre outros parâmetros. A *EnumBenchTool* é responsável também por processar os resultados da etapa de teste que são obtidos pela *dnsperf*.

4.3.3.2 *top*

O comando *top* é uma ferramenta largamente utilizada por usuários Linux para monitorar a utilização dos recursos da máquina para cada processo sendo executado. A *EnumBenchTool* dispara o comando *top* no momento em que a etapa de testes se inicia. Ao receber as instruções enviadas pela *EnumBenchTool*, o comando *top* amostra a utilização de CPU para um dado processo a cada segundo e registra estes dados de medições em um arquivo texto. Posteriormente, este arquivo texto é processado pela ferramenta *EnumBenchTool*.

4.3.3.3 *bwm-ng*

A ferramenta *bwm-ng* [45] oferece ao usuário a possibilidade de monitoração do tráfego de dados que passa por uma dada porta Ethernet. A *EnumBenchTool* utiliza a *bwm-ng* para monitorar o enlace de comunicação utilizado para o envio das consultas e respostas. A ferramenta *bwm-ng* em alguns casos pode utilizar uma considerável parcela do processamento da máquina e, portanto, as medições foram realizadas a partir da máquina cliente, onde não havia restrições quanto a disponibilidade de recursos de processamento. Além disso, ao realizar as medições a partir da máquina cliente, é possível assegurar que não ocorrerão interferências na avaliação do desempenho do software DNS/ENUM sendo executado no servidor.

4.3.3.4 Configuração de Sessão de Testes

Ao configurar uma sessão de testes, o usuário informa o software que deseja avaliar, o tamanho do banco de dados, o número de zonas DNS, a quantidade de registros NAPTR para cada FQDN consultado, dentre outros parâmetros, por meio de um arquivo de configuração da sessão de testes. Ao receber as configurações do usuário, a *EnumBenchTool* se encarrega de gerar os arquivos de consulta e popular o banco de dados de acordo com o software escolhido. Utilizando comunicação via *socket*, a ferramenta sincroniza o início de cada etapa de teste, disparando a ferramenta de emulação de clientes, bem como as demais ferramentas de monitoração de recursos. Ao final da sessão de testes, a ferramenta *EnumBenchTool* informa ao usuário os dados de desempenho do software DNS/ENUM diante de algumas métricas, nomeadamente, a vazão de respostas às consultas, o tempo médio de resposta e a utilização do processador do servidor sob teste.

4.4 Metodologia de Teste

4.4.1 Layout do Ambiente de Teste

O layout do ambiente de testes é baseado no trabalho apresentado em [20]. A Figura 4.1 ilustra o layout de teste considerado. O Gerenciador de Teste é a entidade responsável por sincronizar a ação das máquinas sob teste. Ele envia comandos às outras duas máquinas para que iniciem cada etapa do teste e o monitoramento dos recursos. As mensagens de controle são enviadas em um enlace exclusivo de 100 Mbps. A máquina responsável por emular os clientes executa o aplicativo *dnsperf* que por sua vez envia as consultas ao servidor DNS/ENUM através de um enlace de 1000 Mbps. O número de clientes emulados é incrementado aos poucos sob o controle do Gerenciador de Teste por meio da ferramenta *EnumBenchTool*.

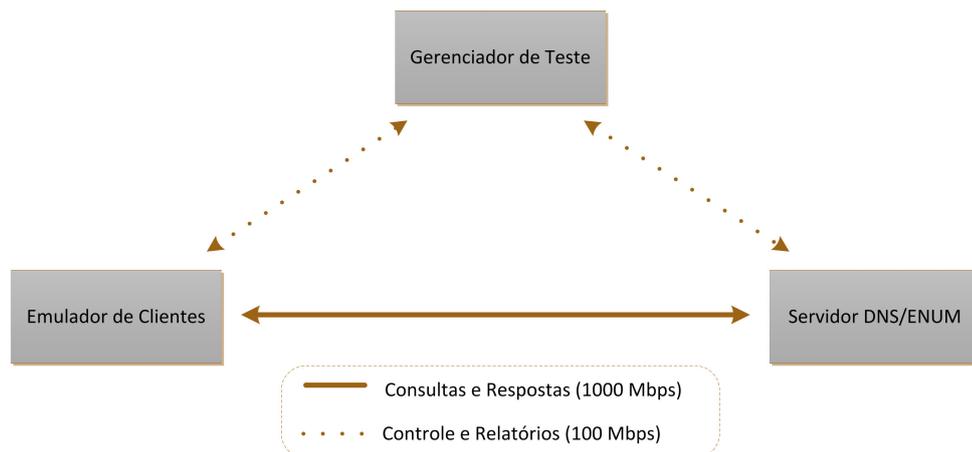


Figura 4.1: Layout do ambiente de testes de avaliação de desempenho de um servidor DNS/ENUM.

4.4.2 Tamanho do Banco de Dados

Com o objetivo de avaliar o desempenho do servidor diante de diferentes tamanhos de banco de dados, foram definidos dois grupos de registros representando diferentes bases de dados. Cada um destes grupos contém 500k e 5M de registros, respectivamente.

4.4.3 Arquivos de Consultas

Como foi descrito na Seção 4.3.3.1, a ferramenta *dnsperf* toma como entrada um arquivo contendo as consultas a serem realizadas. Utilizando a ferramenta *EnumBenchTool*, foram gerados arquivos de consultas para cada um dos grupos de registros definidos. A Tabela 4.1 mostra os intervalos de FQDNs para cada arquivo de consulta.

Tabela 4.1: Intervalos de FQDNs para os arquivos de consultas.

Grupo	Primeiro FQDN	Último FQDN
500k	0.0.0.0.0.3.3.4.3.5.5.e164.arpa	9.9.9.9.4.9.3.3.4.3.5.5.e164.arpa
5M	0.0.0.0.0.0.3.4.3.5.5.e164.arpa	9.9.9.9.4.9.3.4.3.5.5.e164.arpa

4.4.4 Arquivos para Zonas DNS (*Zone Files*)

Utilizando a ferramenta *EnumBenchTool* foram gerados os arquivos para as zonas DNS para cada um dos grupos de registros definidos. Os grupos foram divididos em dez zonas onde cada zona consiste de 1/10 do número total de registros do grupo. A Tabela 4.2 indica a primeira e última zona para cada um dos grupos de registros.

Tabela 4.2: Zonas para cada grupo de registros.

Grupo	Primeira Zona	Última Zona
500k	0.3.3.4.3.5.5.e164.arpa	9.3.3.4.3.5.5.e164.arpa
5M	0.3.4.3.5.5.e164.arpa	9.3.4.3.5.5.e164.arpa

Cada arquivo de zona DNS inicia-se com as diretivas \$TTL e \$ORIGIN. Posteriormente encontra-se o registro SOA, seguido dos registros NS e A para a zona em questão, como ilustrado a seguir:

```
$TTL 86400
```

```
$ORIGIN 0.3.3.4.3.5.5.e164.arpa.
```

```
@ IN SOA ns.0.3.3.4.3.5.5.e164.arpa. root.e164.arpa. (
```

```
2011080901
```

```

21600
3600
604800
3600
)

IN NS ns.0.3.3.4.3.5.5.e164.arpa.
ns IN A 10.0.0.1

```

O restante dos registros no arquivo de zona DNS são todos registros do tipo NAPTR. Cada FQDN do arquivo de consultas possui apenas um registro NAPTR correspondente no arquivo de zonas DNS. Os registros NAPTR armazenados nos arquivos de zonas possuem a seguinte estrutura:

```

0.0.0.0.0 NAPTR 0 0 "U" "sip+E2U" "!^.*$!sip:553433000000@enum.ufu.br!" .
1.0.0.0.0 NAPTR 0 0 "U" "sip+E2U" "!^.*$!sip:553433000001@enum.ufu.br!" .
2.0.0.0.0 NAPTR 0 0 "U" "sip+E2U" "!^.*$!sip:553433000002@enum.ufu.br!" .
3.0.0.0.0 NAPTR 0 0 "U" "sip+E2U" "!^.*$!sip:553433000003@enum.ufu.br!" .
4.0.0.0.0 NAPTR 0 0 "U" "sip+E2U" "!^.*$!sip:553433000004@enum.ufu.br!" .

```

4.5 Resultados dos Testes em Laboratório

Os testes laboratoriais foram constituídos de diversas etapas. Em cada uma destas etapas o número de clientes emulado era incrementado. Dessa maneira, a quantidade de clientes em cada etapa foi controlada pela ferramenta *EnumBenchTool*, que era executada nas máquinas participantes do teste. Cada etapa teve uma duração de 60 segundos, durante os quais a ferramenta *dnsperf* disparou requisições para o servidor DNS/ENUM de acordo com o número de clientes informados pelo gerenciador dos testes. Ao final de cada etapa, o gerenciador aguardou um intervalo de 10 segundos antes de iniciar a etapa seguinte, para que não houvesse sobreposição de resultados entre as etapas de teste.

A Figura 4.2 mostra a vazão obtida para o software BIND, para os dois grupos de registros considerados. É possível observar que o software BIND exibiu um ótimo desempenho com relação a vazão de respostas às consultas, alcançando uma vazão máxima acima de 300 mil consultas por segundo (cps). A Figura 4.3 apresenta o comportamento do tempo médio de resposta para o software BIND. Nota-se que o tempo médio de resposta não

ultrapassou 1 ms. Este baixo tempo de resposta é o principal responsável pela alta vazão proporcionada por este software.

Tanto a Figura 4.2 quanto a Figura 4.3 mostram que o software BIND alcançou a saturação quando havia 300 clientes enviando consultas ao servidor DNS/ENUM. Observe que quando o servidor atinge o seu limite, a vazão de respostas interrompe sua tendência de crescimento e entra em uma fase instável. Da mesma forma, o tempo médio de resposta para de crescer depois do ponto de saturação, enquanto a previsão inicial era que este tempo médio deveria crescer indefinidamente, uma vez que um número cada vez maior de requisições está sendo enviado e o servidor só pode atender parte destas requisições.

A explicação para este comportamento, está na forma como a ferramenta *dnstperf* calcula o tempo médio de resposta. A *dnstperf* só considera as requisições realizadas com sucesso para calcular o tempo médio de resposta e, além disso, não existe um processo de repetição de uma consulta perdida, como ocorre tipicamente em um *resolver*. Dessa forma, mesmo na saturação o servidor responde as requisições com um tempo de resposta próximo daquele encontrado para a vazão máxima e, portanto, o tempo médio de resposta após a saturação exibe uma pequena variação. Contudo, como apenas uma parcela deste crescente número de requisições é atendida, o sistema começa a “experimental” perdas cada vez maiores, conforme ilustrado na Figura 4.4.

A Figura 4.5 ilustra o comportamento do desvio padrão do tempo médio de resposta. Este dados serão importantes para a etapa de modelagem do sistema real.

A Figura 4.6 descreve a utilização da CPU da máquina servidora ao responder às requisições da máquina cliente. Percebe-se que o software BIND não utilizou toda a capacidade de processamento da máquina servidora, confirmando que a saturação do servidor indicada na Figura 4.2 foi causada pelo software e não por restrições de hardware.

Com o objetivo de verificar se outras variáveis não interferiram nos resultados encontrados, duas outras métricas auxiliares foram consideradas:

- Utilização de CPU por parte da ferramenta *dnstperf*: esta métrica auxiliar foi considerada para certificação de que os recursos da máquina cliente foram suficientes para a realização dos testes.
- Utilização do enlace de comunicação: esta métrica verifica como o enlace de comunicação esta sendo utilizado durante o teste. A ferramenta *bwm-ng* monitora na máquina cliente, a taxa gerada de consultas, em bps.

A Figura 4.7 mostra a utilização da CPU na máquina cliente por parte da ferramenta *dnsperf*. É possível observar que as curvas apresentadas estão em concordância com os resultados mostrados nas Figuras 4.2 e 4.3. Além disso, excluindo-se as rotinas típicas do sistema operacional, a ferramenta *dnsperf* era o principal software utilizando os recursos de processamento. Nota-se que apenas uma parcela destes recursos foi consumida, indicando que não houve saturação dos recursos de processamento na máquina cliente, excluindo, portanto, a possibilidade de interferência nos resultados.

A Figura 4.8 mostra a utilização do enlace de comunicação que interconecta o cliente ao servidor. As curvas apresentadas confirmam que o valor médio de utilização ficou abaixo da capacidade máxima do enlace (1000 Mbps), indicando que o enlace não sofreu saturação.

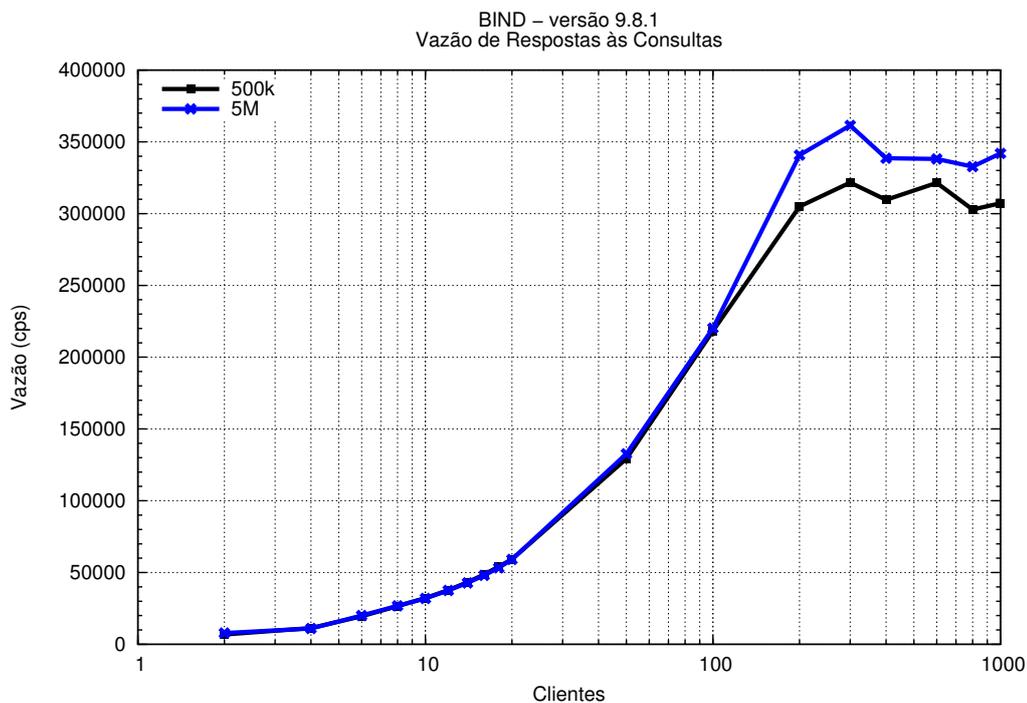


Figura 4.2: Desempenho do servidor DNS/ENUM com relação a vazão de respostas às consultas.

4.6 Considerações Finais

Modelos são abstrações de sistemas reais e devem descrever de forma fidedigna o comportamento do sistema real que representam. Neste sentido, a comparação dos resultados

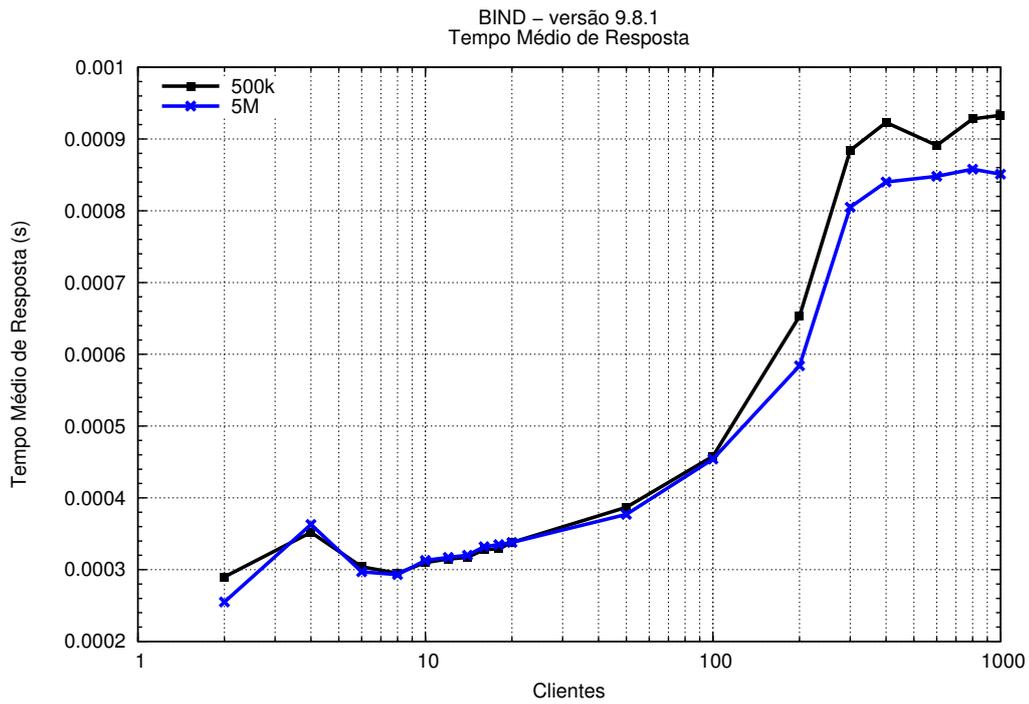


Figura 4.3: Comportamento do tempo médio de resposta.

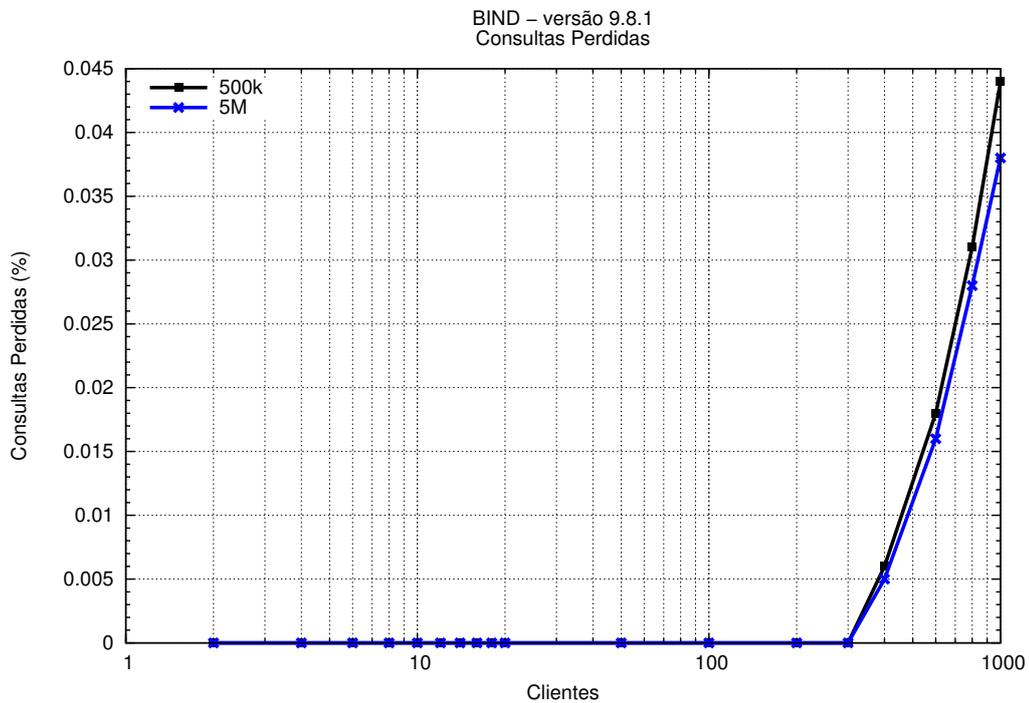


Figura 4.4: Comportamento da taxa de perdas.

obtidos em um sistema real com aqueles obtidos através do modelo é uma das metodologias mais poderosas e utilizadas em processos de validação.

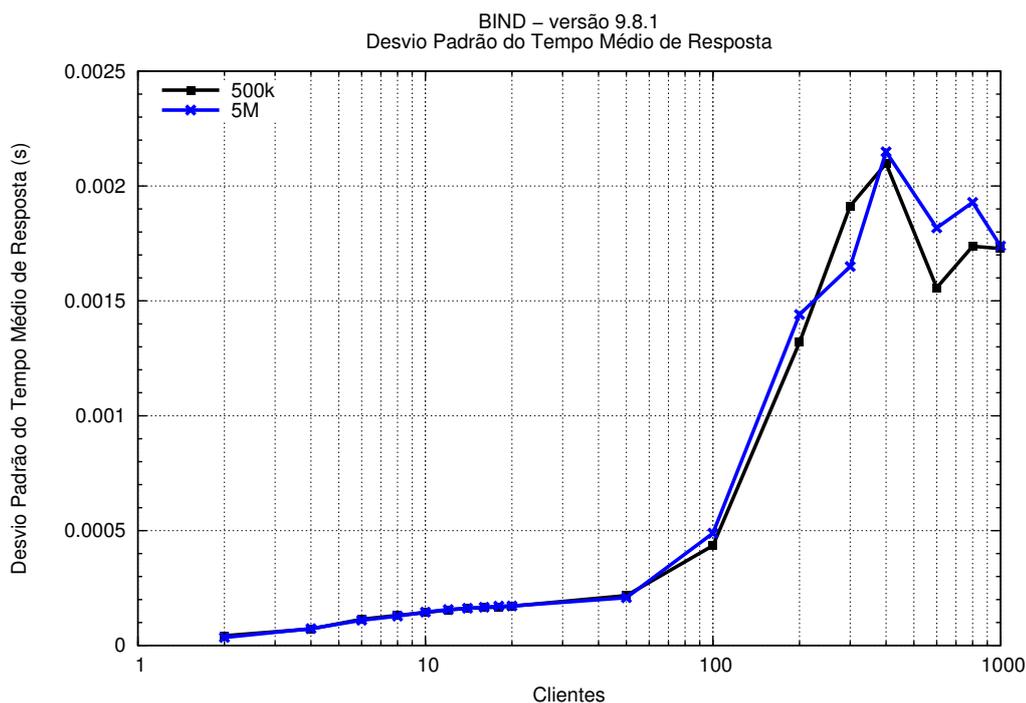


Figura 4.5: Comportamento do desvio padrão do tempo médio de resposta.

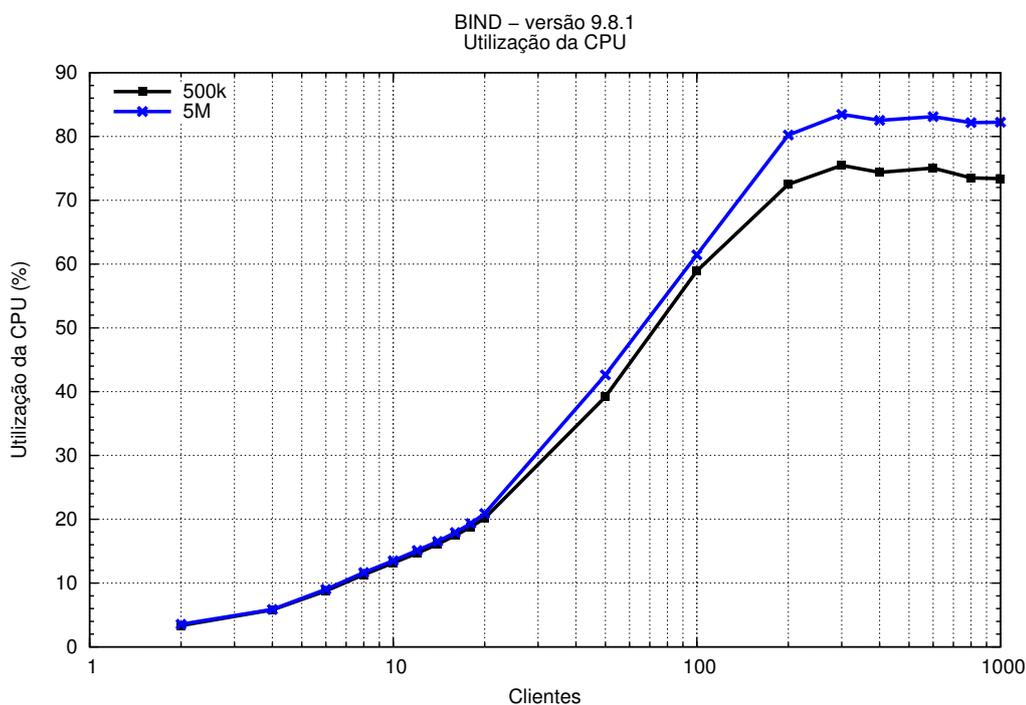


Figura 4.6: Utilização da CPU pelo software BIND.

Este capítulo foi dedicado a descrever os testes realizados em laboratório com o objetivo de obter dados de um sistema real e possibilitar a futura comparação e validação do modelo

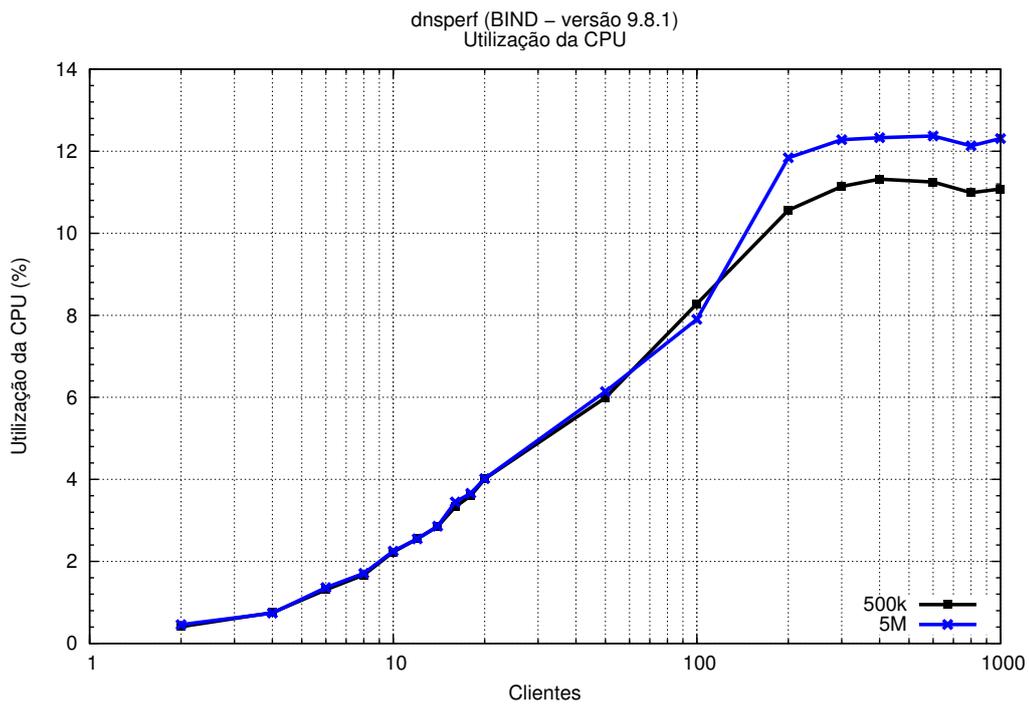


Figura 4.7: Utilização da CPU pela *dnsp erf*.

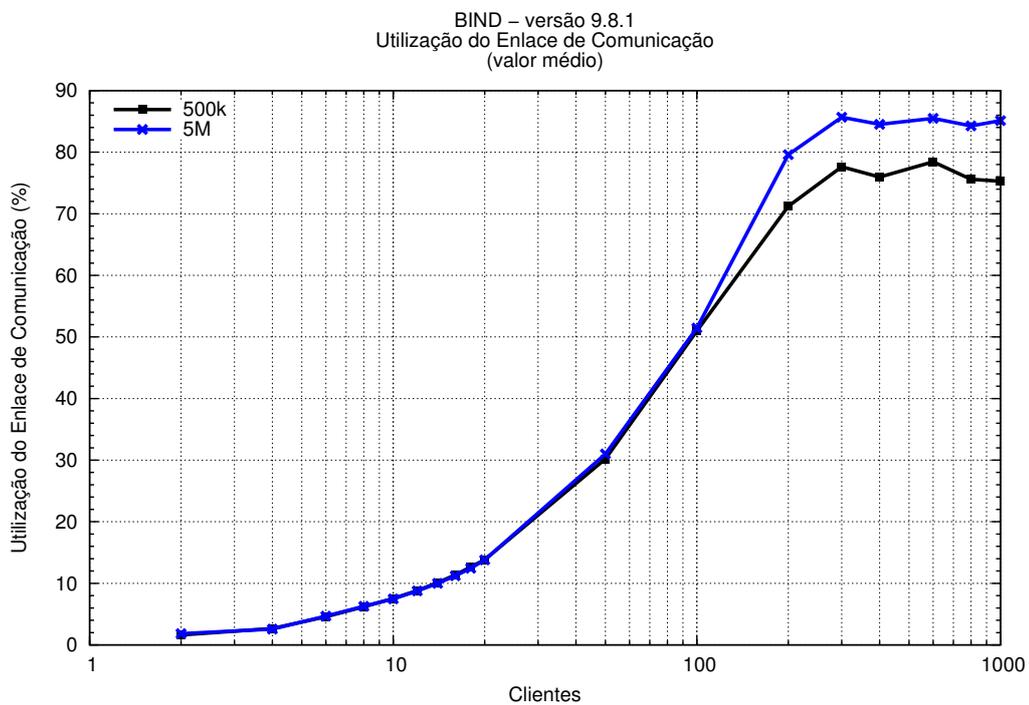


Figura 4.8: Utilização do enlace de comunicação.

de simulação proposto nesta dissertação.

Os resultados comprovam que o software BIND apresentou alto desempenho quanto

a vazão de respostas às consultas e ao tempo médio de resposta. Este software também exibiu um ótimo desempenho quanto a escalabilidade do banco de dados. Os resultados demonstram ainda que a vazão alcançada por este software está bem acima daquela apresentada em [19], devido principalmente aos investimentos em hardware e software realizados nos últimos anos.

No próximo capítulo, é proposto um modelo matemático para a descrição do comportamento do servidor DNS/ENUM, baseando-se nos resultados obtidos em laboratório.

Capítulo 5

Descrição do Modelo de Simulação

5.1 Introdução

A plataforma de simulação é composta por módulos de simulação que representam cada um dos elementos presentes em um sistema ENUM, tais como os clientes e os servidores de nomes autoritativos. Este capítulo apresenta o modelo matemático proposto para a abstração de cada um destes elementos reais. O estudo é iniciado pela ferramenta de emulação de clientes. Em seguida encontram-se as considerações sobre o modelo adotado para descrever o comportamento do servidor DNS/ENUM de acordo com as curvas de desempenho apresentadas no Capítulo 4. O capítulo é finalizado com os comentários acerca do modelo para o enlace de comunicação que interconecta os elementos do sistema.

5.2 Modelo para a ferramenta *dnstperf*

Como mencionado no Capítulo 4, a ferramenta *dnstperf* foi criada pela empresa Nominum. Esta ferramenta é amplamente utilizada por pesquisadores interessados em avaliar o desempenho de um servidor DNS autoritativo. Além de ser uma ferramenta com muitas funcionalidades, a *dnstperf* é um software *open-source*, o que permite que os pesquisadores possam realizar pequenas modificações de acordo com as particularidades de seus testes.

Nos testes descritos no Capítulo 4, nem todas as funcionalidades da ferramenta *dnstperf* foram utilizadas (DNSSEC, atualização de registros, etc). Logo, o modelo proposto para esta ferramenta foi re-escrever as funcionalidades utilizadas nos testes de avaliação para serem integradas ao ambiente de simulação NS-3. Analisando o código original da fer-

ramenta *dnperf* escrito na linguagem C, foi possível propor um módulo de simulação similar, escrito em C++, para ser integrado ao NS-3. Os detalhes desta implementação são apresentados no Capítulo 6.

5.3 Modelagem dos Servidores Autoritativos

Um servidor de nomes autoritativo é aquele responsável pelo armazenamento dos registros de recursos referentes aos domínios que possui autoridade. No Capítulo 4 foram apresentados dados de desempenho de um servidor de nomes autoritativo atuando em um contexto ENUM. O modelo proposto para este servidor será baseado na análise gráfica das curvas de desempenho. Esta análise gráfica permite avaliar o comportamento do servidor diante de valores de tráfego distintos. Observando as Figuras de 4.2 a 4.8 é possível verificar que, na maioria dos casos, o comportamento do servidor não é dado por uma única função, mas, na verdade, é o resultado da combinação de diferentes funções, de acordo com o número de clientes emulados. Diante desta constatação, o procedimento para a modelagem deste servidor seguiu uma modelagem híbrida, isto é, a curva original é segmentada e cada um destes trechos é aproximado por uma função adequada.

5.3.1 Considerações sobre a Modelagem

Como dito anteriormente, a análise gráfica dos dados provenientes das medições realizadas nos testes em laboratório indicou um comportamento híbrido para o servidor de nomes analisado. Esta constatação é reportada em [2] onde o autor realizou medições no sistema ENUM francês e observou que o comportamento dos servidores poderia ser dividido em duas fases: uma fase inicial aproximada para uma função exponencial e uma fase final, aproximada para uma função linear.

A identificação de que o comportamento do servidor é resultado da combinação de mais de uma expressão matemática implica na necessidade de determinar um ponto de transição entre os distintos trechos que compõem a curva de desempenho. Dessa maneira, para determinar este ponto de transição, a recomendação dada em [2] é utilizar o valor de vazão máxima registrado para o servidor em questão. Entretanto, em alguns casos pode ser mais interessante segmentar a curva em outro ponto e, dessa forma, percebe-se que uma análise generalista pode não ser adequada dada as diferenças de comportamento

destas curvas. Observa-se que na Figura 4.2, a primeira parte da curva é similar a uma função exponencial, enquanto que a segunda parte (pós-saturação) é mais linear. Por outro lado, na Figura 4.4, os dois segmentos estão mais próximos de uma função linear. Partindo deste pressuposto, esta dissertação propõe uma análise mais detalhada das curvas que descrevem o comportamento do servidor de nomes, em contraste com a análise mais generalista apresentada em [2].

Serão propostas e avaliadas duas estratégias de modelagem para o servidor de nomes autoritativo. A primeira abordagem é uma extensão da metodologia apresentada em [2] baseada em regressão de funções. A segunda estratégia é baseada na utilização das amostras obtidas nos testes do sistema real.

5.3.2 Modelagem Baseada em Regressão de Funções

Nesta seção são apresentados os procedimentos utilizados para aproximar as curvas de comportamento do servidor de nomes real em funções matemáticas que serão utilizadas no simulador. O estudo é iniciado pela regressão linear que é mais simples e, posteriormente, a regressão exponencial é apresentada.

5.3.2.1 Regressão Linear

A regressão linear é utilizada para estudar as relações entre diferentes variáveis com o propósito de realizar predições. Denota-se por y a variável aleatória a ser predita (também conhecida como variável dependente) e denota-se por x a variável independente (ou preditora) utilizada para modelar y [46].

A forma mais simples de modelos de regressão é conhecida como regressão linear simples. Neste tipo de regressão tem-se apenas uma variável dependente y e uma única variável independente x . Logo, é possível definir um modelo de regressão linear simples por,

$$y = \beta_0 + \beta_1 x + \varepsilon \quad (5.1)$$

onde β_0 é o interceptor do eixo y , β_1 é a inclinação da reta e ε é uma variável aleatória que representa o erro na predição da variável dependente. É necessário, então, encontrar os valores dos estimadores para β_0 e β_1 . Ao encontrar estes estimadores ($\hat{\beta}_0$ e $\hat{\beta}_1$) pode-se traçar uma reta dada pela equação de predição 5.2:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x. \quad (5.2)$$

Portanto, o cerne da regressão linear se concentra na determinação dos valores dos estimadores que melhor descrevam os dados a partir da equação de predição. Dentre os métodos que permitem encontrar estes estimadores, o mais utilizado é o Método dos Mínimos Quadrados, que será empregado nesta dissertação. Tal metodologia é apresentada no Apêndice B. Ao obter-se os valores para os estimadores $\hat{\beta}_0$ e $\hat{\beta}_1$, a equação estimadora é determinada.

5.3.2.2 Regressão Exponencial

Quando o estudo de regressão é realizado, nem sempre é possível aproximar os dados observados para uma função linear. Neste caso, tem-se um modelo não-linear e mais complexo que o anterior. A regressão exponencial é uma das ferramentas utilizadas para transformar um modelo não-linear em um modelo linear. É possível identificar dois tipos de modelos não-lineares [47]:

- modelos intrinsecamente lineares: $y = \theta_0 x^{\theta_1} \epsilon$
- modelos intrinsecamente não-lineares: $y = \theta_0 x^{\theta_1} + \epsilon$

onde ϵ é uma variável aleatória que representa o erro na predição da variável dependente. Dentre estes dois modelos, o intrinsecamente linear pode ser transformado em um modelo linear simples, como o estudado na Seção 5.3.2.1.

Nesta dissertação será utilizado um modelo exponencial que é do tipo intrinsecamente linear. Neste contexto, considerando o modelo exponencial $y = \theta_0 e^{\theta_1 x} \epsilon$, pode-se convertê-lo em um modelo linear simples aplicando-se o logaritmo natural em ambos os lados da equação, obtendo 5.3:

$$\begin{aligned} \ln(y) &= \ln(\theta_0 e^{\theta_1 x} \epsilon) \\ \ln(y) &= \ln(\theta_0) + \theta_1 x + \ln(\epsilon) \\ y' &= \beta_0 + \beta_1 x + \epsilon' \end{aligned} \quad (5.3)$$

onde: $y' = \ln(y)$, $\beta_0 = \ln(\theta_0)$, $\beta_1 = \theta_1$, $\epsilon' = \ln(\epsilon)$.

Logo, a partir da equação linear obtida em 5.3, aplica-se o procedimento de regressão descrito na Seção 5.3.2.1. Ao encontrar os estimadores β_0 e β_1 é possível determinar θ_0 e θ_1 e, conseqüentemente, a equação de predição.

5.3.2.3 Qualidade da Regressão

Determinar os valores dos estimadores não é o suficiente para assegurar que o modelo é suficientemente adequado para descrever os dados observados. Nesse sentido, a qualidade da regressão, ou seja, o quanto o modelo é próximo da situação real, pode ser determinado por meio de algumas ferramentas estatísticas.

Para a regressão linear, a métrica mais utilizada é o coeficiente de determinação, denotado por R^2 , o qual indica o quanto a regressão conseguiu acompanhar a variação da variável dependente. O valor do coeficiente de determinação está compreendido no intervalo $0 \leq R^2 \leq 1$, sendo que a qualidade da regressão é melhorada à medida que o R^2 tende à unidade. No Apêndice B, o leitor pode encontrar detalhes de como determinar o valor de R^2 .

Uma outra importante métrica utilizada é o coeficiente de correlação, denotado por r . Tal parâmetro indica o quanto duas variáveis aleatórias são linearmente dependentes, podendo assumir valores que se encontram entre -1 e 1 . Dessa maneira, à medida que r tende a -1 ou 1 , verifica-se que, mais forte é a relação linear entre as variáveis analisadas. Quando existe apenas uma única variável independente é possível estabelecer que $r^2 = R^2$.

Para a regressão exponencial, a transformação do modelo exponencial em um modelo linear causa um distúrbio na escala dos dados e a utilização do coeficiente de determinação R^2 como métrica de desempenho da regressão não é confiável. Em [47] existe a recomendação do uso do quadrado do coeficiente de correlação (r^2) entre os valores reais e os valores estimados transformados de volta para a escala original. Detalhes de como determinar o valor de r são dados no Apêndice B.

Além dessas duas métricas, é importante determinar o intervalo de confiança para os valores do intercepto (β_0) e a inclinação da reta (β_1). O Apêndice B descreve o procedimento utilizado para obtenção destes intervalos.

5.3.2.4 Identificação de *Outliers*

Para realizar o processo de regressão, um importante passo é identificar os melhores pontos na curva, que possibilitam a obtenção de uma função matemática mais próxima da curva real. Algumas curvas podem apresentar pontos com comportamento diferente do padrão apresentado pelas demais amostras. Geralmente estes pontos são conhecidos como *outliers* [46], os quais são, aconselhadamente, removidos para se obter um modelo que descreva os dados com maior fidelidade.

Nas Figuras de 4.2 a 4.8, foram amostrados apenas alguns pontos para apresentar o comportamento geral do software. Com o objetivo de conhecer melhor a variação do comportamento do software em razão do número de clientes no sistema, ampliou-se o número de amostras avaliadas. As Figuras 5.1, 5.2 e 5.3 ilustram este cenário em que foram utilizadas 1000 amostras.

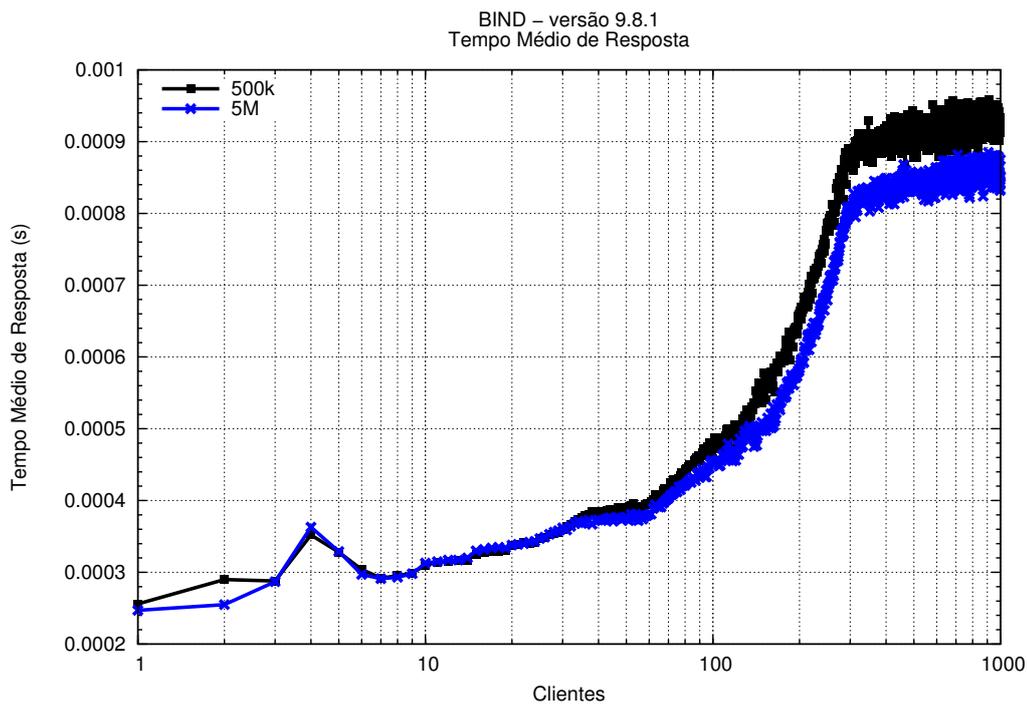


Figura 5.1: Comportamento do tempo médio de resposta (1000 amostras).

Nestas duas figuras é possível notar que o comportamento do servidor não é totalmente uniforme de acordo com o aumento do número de clientes. Esta variação é mais intensa ainda quando o servidor está na zona de saturação. Dessa forma, é necessário aplicar estratégias de remoção de *outliers*. A identificação de *outliers* pode ser realizada de

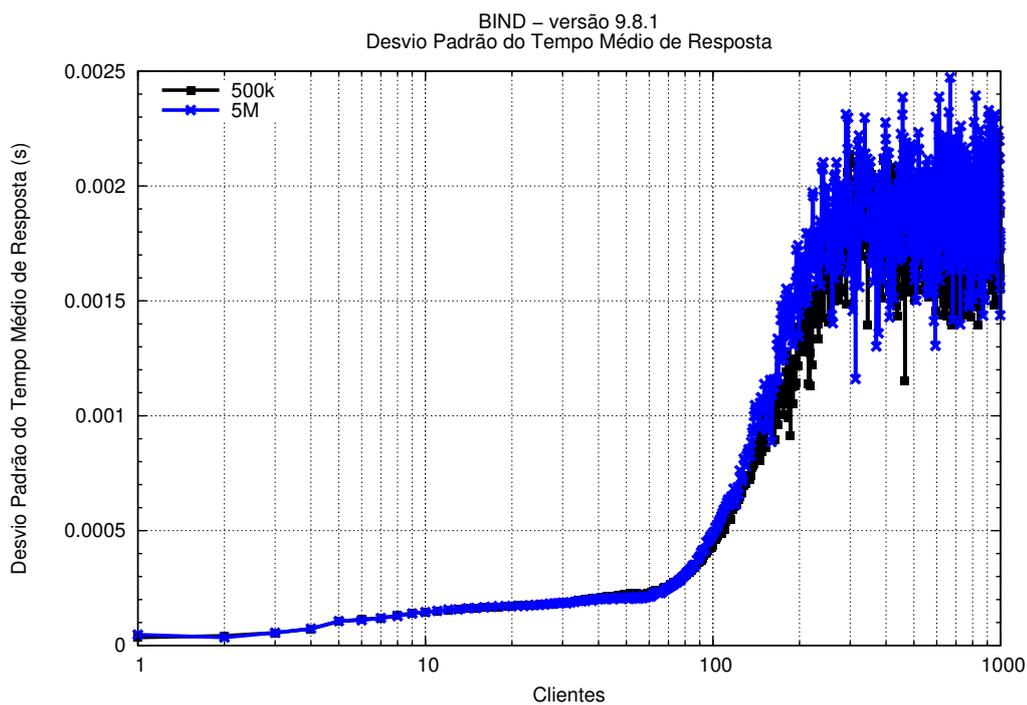


Figura 5.2: Comportamento do desvio padrão do tempo médio de resposta (1000 amostras).

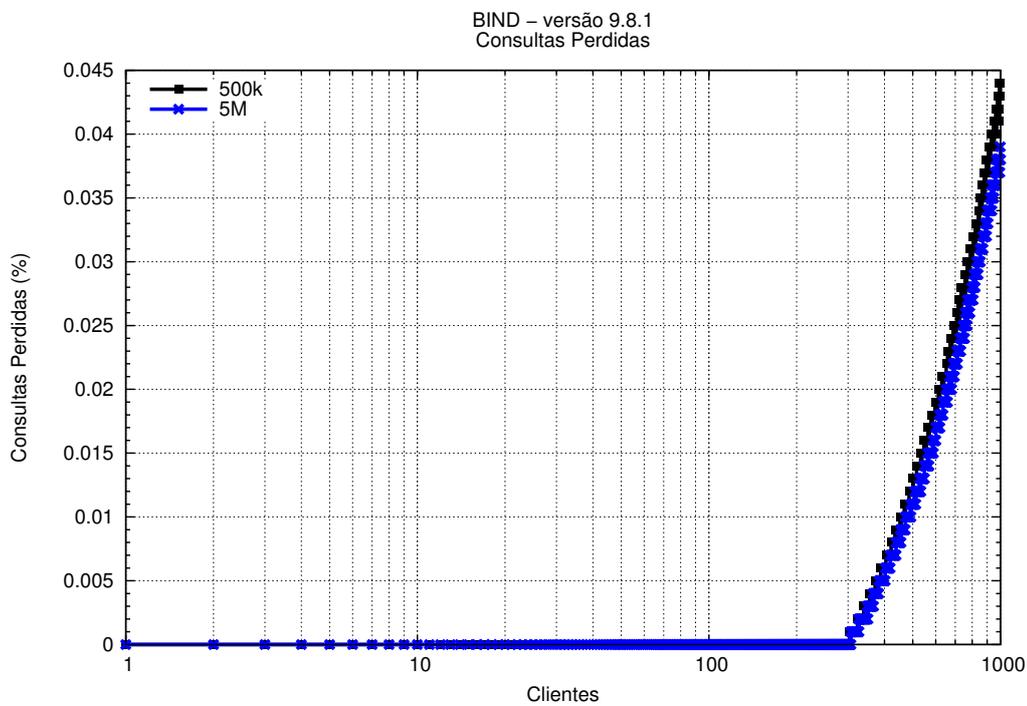


Figura 5.3: Comportamento da taxa de perdas (1000 amostras).

duas formas: (1) por meio da análise gráfica dos dados em um gráfico *scatterplot* que se caracteriza como um diagrama com duas variáveis em coordenadas Cartesianas; (2) por meio de análise estatística dos dados em questão. Nesta dissertação estas duas abordagens foram utilizadas, conjuntamente, a fim de identificar possíveis *outliers*.

Dessa maneira, a análise gráfica consistiu na identificação imediata de pontos distantes da reta através da visualização dos dados em gráficos, enquanto que a avaliação estatística foi realizada através do teste *z-score* modificado [46], apresentado no Apêndice B.

O resultado deste estudo pode ser visualizado nas Figuras 5.4, 5.5 e 5.6. Observa-se que a curva que representa os dados "filtrados", isto é, com tratamento para a remoção de *outliers* é bastante uniforme, o que facilita o processo de regressão. A exceção é a zona de saturação que, por natureza, é caracterizada por um comportamento instável.

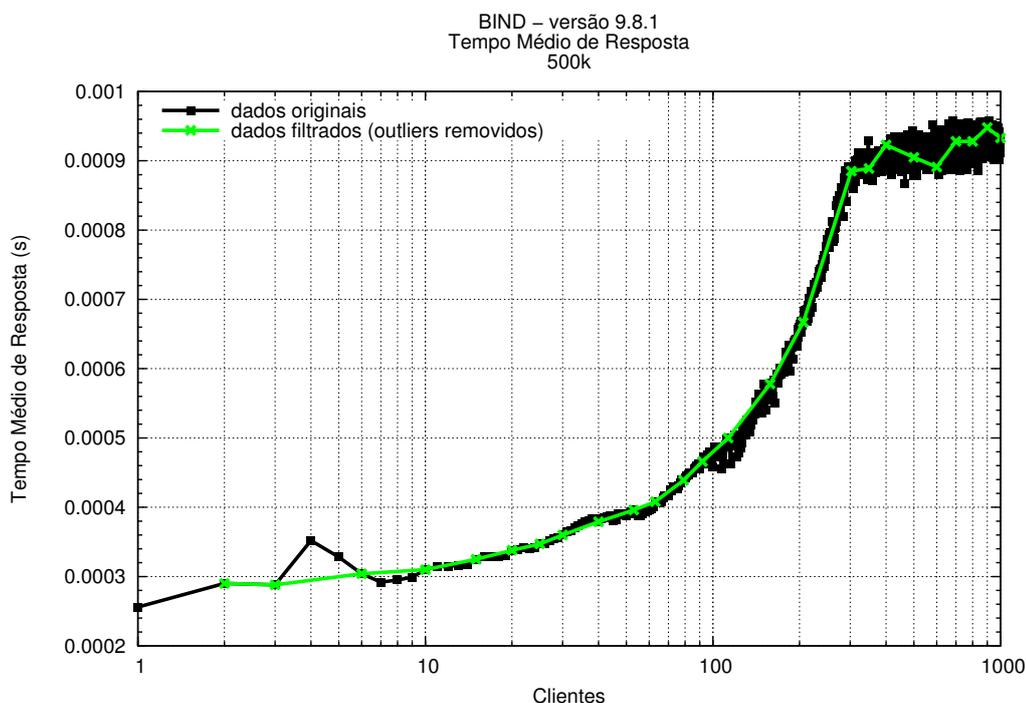


Figura 5.4: Comportamento do tempo médio de resposta (*outliers* removidos).

5.3.2.5 Equações de Predição

Utilizando os conceitos apresentados nas seções anteriores, foi possível determinar as equações de predição para o servidor analisado. Para cada equação foi realizada a análise estatística, para verificar a qualidade da equação na predição dos valores reais. A Tabela 5.1 apresenta os resultados da modelagem realizada. A variável x representa o número de

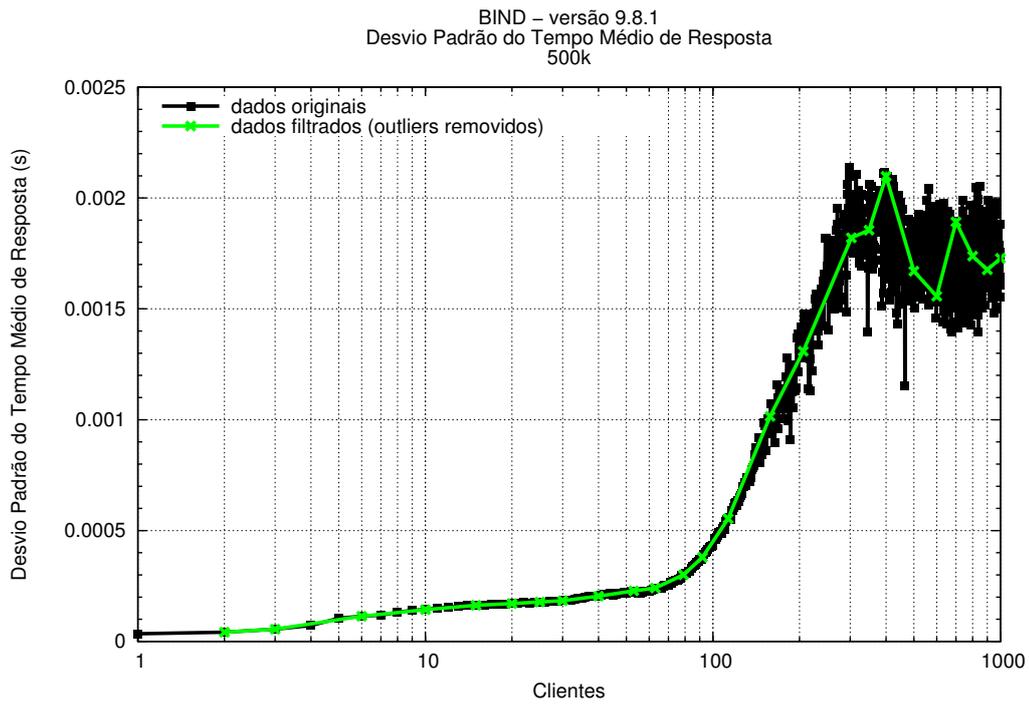


Figura 5.5: Comportamento do desvio padrão do tempo médio de resposta (*outliers* removidos).

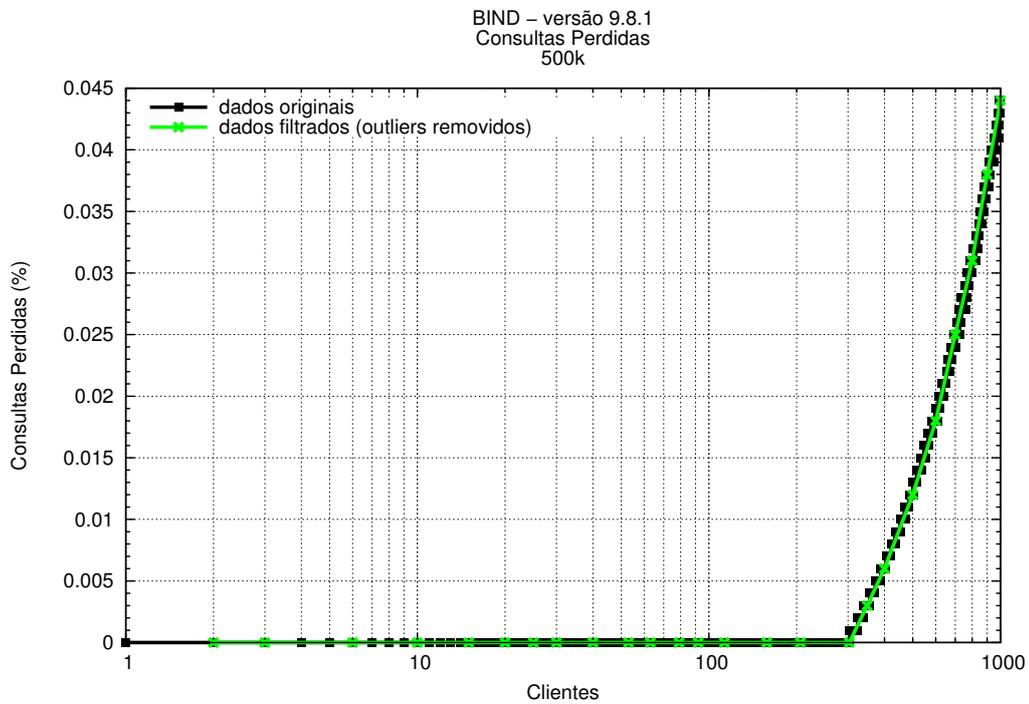


Figura 5.6: Comportamento da taxa de perdas (*outliers* removidos).

clientes emulados. A variável y é compartilhada pelas métricas tempo médio de resposta, desvio padrão do tempo médio de resposta e taxa de perdas, e em cada tabela existe a indicação de qual dessas métricas a variável y está denotando.

Na Tabela 5.1, o valor de r foi calculado para diferentes pares de variáveis conforme descrito nas Seções 5.3.2.1 e 5.3.2.2. Para a regressão linear o valor do coeficiente de correlação (r) foi calculado para as variáveis x e y . Para a regressão exponencial, o valor de r foi determinado para as variáveis x e $\ln(y)$ devido à transformação realizada para linearizar os dados. Como as equações possuem apenas uma variável independente é possível dizer que $r^2 = R^2$ e, portanto, as tabelas apresentam o valor de r_{xy}^2 para a regressão linear. Para o caso da regressão exponencial, o valor de r^2 foi calculado com relação às variáveis y e \hat{y}_{raw} , onde \hat{y}_{raw} representa os valores estimados para a variável y transformados de volta à escala original. O intervalo de confiança foi calculado para todos os estimadores com 95% de confiança de que os valores verdadeiros estão nos intervalos apresentados. O intervalo de atuação estabelece o intervalo no qual a função é considerada para modelar o servidor, indicando claramente o limiar de transição entre as funções.

Observa-se que na maioria dos casos os valores de r e r^2 mantiveram-se acima de 0.9, sinalizando que a modelagem proposta é adequada para representar o comportamento real dos servidores. Entretanto, a função modelo para o intervalo em que o servidor está saturado não obteve coeficientes estatísticos satisfatórios devido à grande instabilidade que o servidor apresenta nesta fase, fato este que dificulta a aproximação por uma única função. Neste sentido, a próxima seção propõe uma solução para este problema.

Além da análise estatística, a adequação do modelo pode ser visualizada nas Figuras 5.7, 5.8 e 5.9, que ilustram a comparação entre a curva real e o modelo proposto.

5.3.3 Modelagem Direta Baseada em Amostras

A Tabela 5.1 apresenta os resultados do processo de modelagem do servidor de nomes autoritativo baseada na regressão de funções. Apesar da maioria dos coeficientes indicarem que as regressões conseguem descrever de forma satisfatória o comportamento da curva real, ainda existem alguns segmentos desta curva real que não foram aproximados satisfatoriamente pela regressão.

Tabela 5.1: Equações de predição para o servidor DNS/ENUM considerando-se 500k registros.

Tempo Médio de Resposta				
Intervalo de Atuação	Equação de Predição	r	r^2	Intervalo de Confiança (95%)
$0 \leq x < 303$	$y = 0.000311e^{0.003716x}$	0.98653 ¹	0.97917 ³	$\theta_0: (0.000301, 0.000323)$ $\theta_1: (0.003776, 0.004055)$
$303 \leq x < \infty$	$y = 0.000870 + 7.20 * 10^{-8}x$	0.79791 ²	0.63667 ²	$\beta_0: (0.000837, 0.000902)$ $\beta_1: (2.34 * 10^{-8}, 1.21 * 10^{-7})$
Desvio Padrão do Tempo Médio de Resposta				
Intervalo de Atuação	Equação de Predição	r	r^2	Intervalo de Confiança (95%)
$0 \leq x < 79$	$y = 0.000089 + 0.000002x$	0.92946 ²	0.86390 ²	$\beta_0: (0.000061, 0.000118)$ $\beta_1: (0.000002, 0.000003)$
$79 \leq x < 303$	$y = -0.000207 + 0.000007x$	0.99047 ²	0.98104 ²	$\beta_0: (-0.000444, 0.000030)$ $\beta_1: (0.000006, -0.000008)$
$303 \leq x < \infty$	$y = 0.001946 - 2.66 * 10^{-7}x$	-0.42247 ²	0.17848 ²	$\beta_0: (0.001608, 0.002282)$ $\beta_1: (-7.77 * 10^{-7}, 2.44 * 10^{-7})$
Taxa de Perdas				
Intervalo de Atuação	Equação de Predição	r	r^2	Intervalo de Confiança (95%)
$0 \leq x < 303$	$y = 0$	-	-	-
$303 \leq x < \infty$	$y = -0.018875 + 6.26 * 10^{-5}x$	0.99952 ²	0.99903 ²	$\beta_0: (-0.020025, -0.017724)$ $\beta_1: (6.09 * 10^{-5}, 6.44 * 10^{-5})$

¹ $r_{x \ln y}$

² r_{xy}

³ $r_{y \hat{y}_{raw}}$

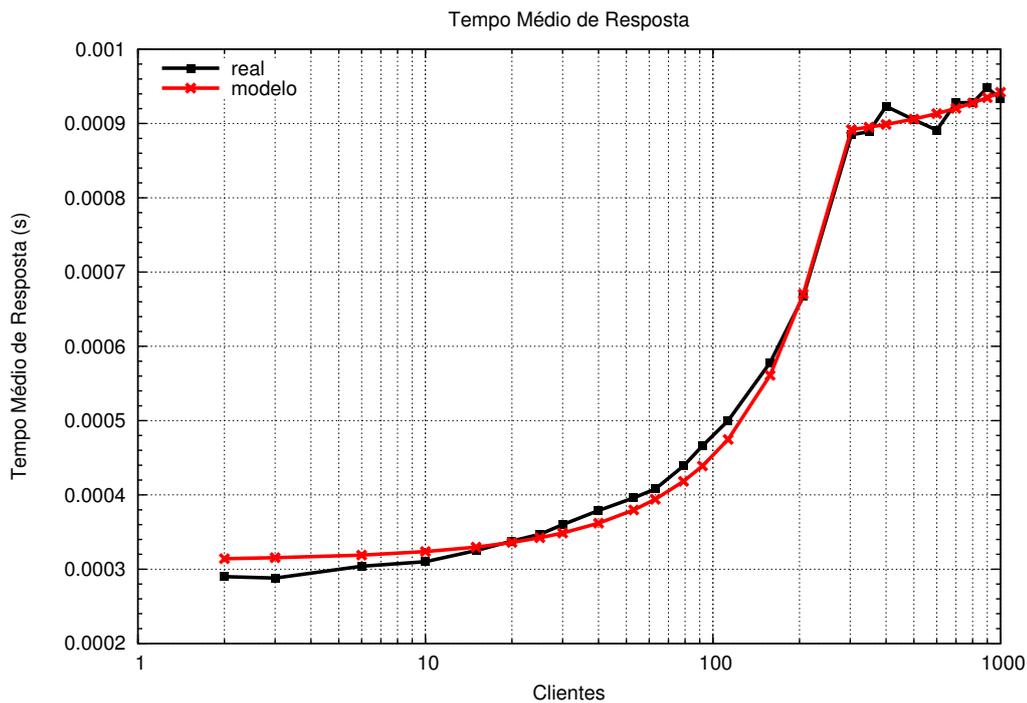


Figura 5.7: Comparação entre comportamento real e o modelo proposto para o tempo médio de resposta, considerando-se 500k registros no banco de dados.

Esta seção descreve uma segunda abordagem para a modelagem do servidor de nomes com o objetivo de oferecer uma alternativa à regressão de funções nas situações em que esta não descreve de forma satisfatória o comportamento real do servidor. De forma semelhante a regressão de funções, esta segunda estratégia de modelagem também é baseada na curva real obtida nos testes em laboratório. Entretanto, esta modelagem utiliza as próprias amostras para descrever o comportamento do servidor, sem a necessidade de realizar aproximações.

Considere as tabelas apresentadas no Apêndice A. Para um dado valor amostrado da variável independente x (por exemplo, $x_i = 20$ clientes), o valor da variável dependente y no modelo será o próprio valor real $y_i = 338\mu s$.

Quando o modelo precisa encontrar o valor da variável y para um valor de x que não foi amostrado (por exemplo, $x' = 22$ clientes), procede-se a realização de uma interpolação entre os pontos mais próximos, ou seja:

- O modelo encontra as duas amostras mais próximas de $x' = 22$:

$$x_i = 20$$

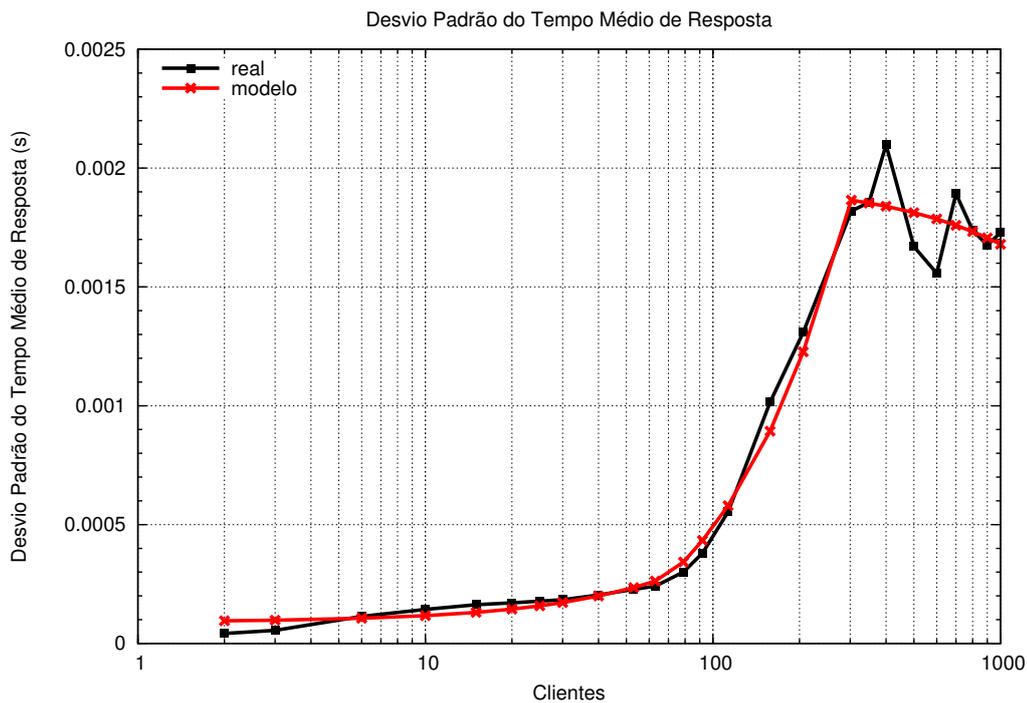


Figura 5.8: Comparação entre comportamento real e o modelo proposto para o desvio padrão do tempo médio de resposta, considerando-se 500k registros no banco de dados.

$$x_{i+1} = 25$$

- A seguir, o modelo encontra a equação da reta que passa pelos pontos (x_i, y_i) e (x_{i+1}, y_{i+1}) :

$$y = 1.8x + 302$$

- Utilizando a equação encontrada no passo anterior, o modelo calcula o valor de y' quando $x' = 22$:

$$y' = 341.6\mu s$$

Esta abordagem oferece maior acurácia que o modelo baseado em regressão linear, uma vez que utiliza a própria curva real. Quanto maior o número de amostras, maior é a precisão desta estratégia. Logo, para esta abordagem é melhor utilizar os dados coletados com 1000 amostras, ao invés dos dados "filtrados", utilizados no processo de regressão.

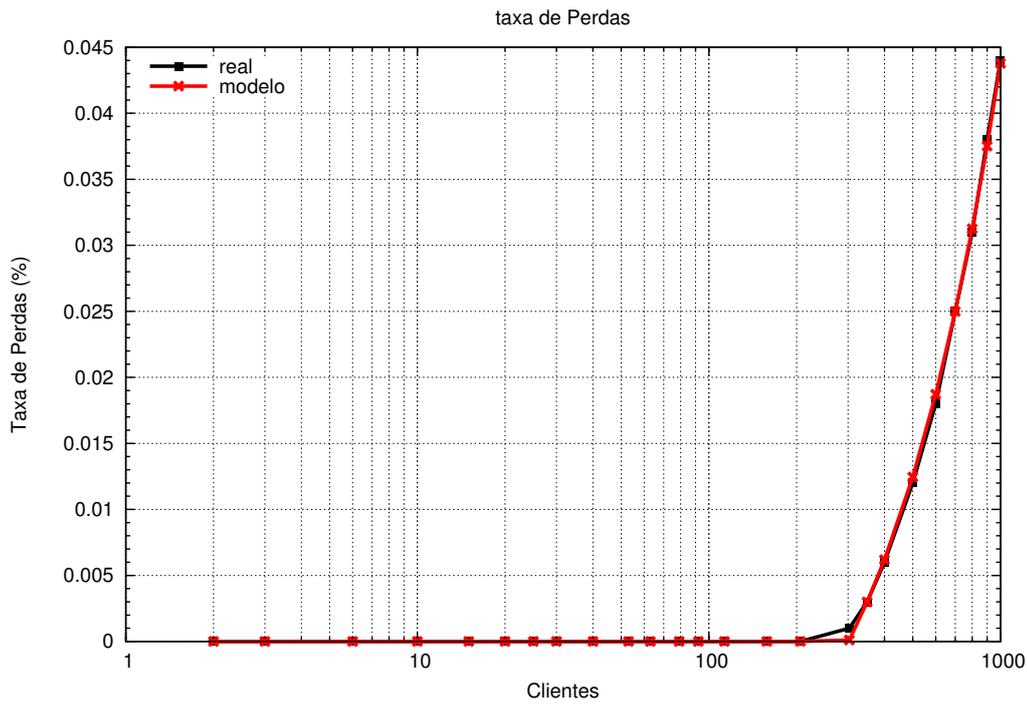


Figura 5.9: Comparação entre comportamento real e o modelo proposto para a taxa de perdas, considerando-se 500k registros no banco de dados.

5.4 Modelagem do Enlace de Comunicação

O software NS-3 possui um modelo simples para os enlaces que interconectam os nós da rede. Tal modelo possibilita a configuração do enlace quanto a largura de banda, atraso de propagação e tipo de fila do *buffer*. Apesar de simples, este modelo é satisfatório para o estudo realizado nesta dissertação.

O cabo utilizado nos experimentos em laboratório para interligar as máquinas de teste ao *switch* foi um cabo Ethernet Furukawa Patch Cord U/UTP GigaLan Cat.6. Segundo [48], a velocidade de propagação nominal no cabo é 66% da velocidade de propagação da luz. Sabendo-se que foram utilizados aproximadamente 5 metros deste cabo, é possível calcular o atraso de propagação, através da equação 5.4:

$$d_{prop} = \frac{\text{comprimento} - \text{do} - \text{enlace}}{v_{luz} * 0.66}$$

$$d_{prop} = \frac{5}{3 * 10^8 * 0.66}$$

$$d_{prop} = 25ns \tag{5.4}$$

Sendo assim o modelo dos enlaces considera uma largura de banda de 1000Mbps e um atraso de propagação de 25ns.

5.5 Considerações Finais

Este capítulo foi dedicado à descrição do processo de modelagem do sistema real estudado no Capítulo 4. Dentro deste propósito, a ferramenta *dnsperf* foi analisada para construir uma ferramenta análoga com funcionalidades mais reduzidas de acordo com o contexto dos testes realizados em laboratório. O processo de implementação desta ferramenta e integração ao NS-3 é apresentado no Capítulo 6.

Para a modelagem do servidor de nomes autoritativo foram propostas duas estratégias: modelagem baseada na regressão de funções e modelagem direta baseada em amostras e equação da reta.

A modelagem baseada na regressão de funções foi proposta no trabalho apresentado em [2]. Este capítulo propõe uma extensão a esta abordagem, cujo objetivo é uma melhor aproximação dos dados reais, bem como o preenchimento de algumas lacunas do processo de modelagem apresentado em [2], tais como a ausência de definição clara do intervalo de atuação de cada equação de predição.

A segunda estratégia, que baseia-se na utilização direta das amostras obtidas nos testes em laboratório, promete um desempenho superior ao oferecido pela modelagem baseada na regressão de funções. Estas duas abordagens são testadas na plataforma de simulação e avaliadas no Capítulo 6 para a determinação de qual delas oferece um modelo mais fiel ao sistema real em estudo.

Por fim, este capítulo define o modelo de enlace simples utilizado no ambiente de simulação NS-3 como o modelo que será aplicado aos enlaces presentes na plataforma de simulação.

O próximo capítulo é dedicado à apresentação do processo de implementação, integração e validação da plataforma de simulação.

Capítulo 6

Implementação da Plataforma de Simulação

6.1 Introdução

No Capítulo 4 foram apresentadas as medições realizadas em laboratório. A partir destas medições foram propostos modelos matemáticos para descrever o sistema real considerado, os quais foram apresentados no Capítulo 5. O presente capítulo pretende descrever o processo de implementação dos modelos propostos. Neste contexto, este capítulo é destinado também à apresentação do ambiente de simulação NS-3, além de descrever como foi realizada a integração da plataforma de simulação ao NS-3. Por fim, o capítulo é concluído com a apresentação e validação dos resultados obtidos via simulação.

6.2 Considerações de Projeto do Simulador

Um modelo de simulação nem sempre precisa reproduzir um sistema real considerando todos os seus detalhes e complexidade, visto que algumas hipóteses podem ser consideradas para tornar este modelo mais simples. Contudo, um modelo detalhado geralmente oferece melhores resultados, uma vez que realiza poucas suposições e, conseqüentemente, diminui-se o erro entre os resultados obtidos no sistema real e aqueles encontrados através do modelo de simulação [2].

Para o modelo de simulação apresentado neste capítulo preferiu-se uma abordagem detalhista. Esta escolha foi realizada tendo como princípios as seguintes considerações:

- Ao invés de considerar um sistema complexo e construir um modelo de simulação com diversas suposições, o projeto do modelo de simulação considerou um sistema simples, em que todos os parâmetros são conhecidos. Dessa forma, o modelo de simulação é detalhado, reproduzindo o sistema real com poucas suposições.
- Definir um sistema real simples possibilitou a realização de medições neste sistema e obter dados reais que poderiam ser utilizados na validação do modelo.
- As métricas de avaliação são conhecidas: vazão de respostas às consultas e tempo médio de resposta.
- O protocolo em estudo é baseado no DNS que é um assunto já consolidado, com vasta documentação que permite ao pesquisador o acesso a detalhes do sistema.

Diante destas considerações foi possível implementar um modelo de simulação detalhado. A construção da plataforma de simulação considerou também a possibilidade de inclusão futura de novos modelos de simulação para sistemas reais mais complexos do que o considerado neste trabalho.

6.2.1 Ambiente de Implementação

Como foi discutido na Seção 3.3.3, o ambiente de simulação escolhido para a implementação da plataforma de simulação foi o *Network Simulator 3* (NS-3).

O NS-3 é um simulador de rede do tipo *discrete-event* em que o núcleo e os modelos do simulador são implementados em C++ [49] [37]. É importante ressaltar que o NS-3 não é uma extensão do NS-2, mas um novo simulador totalmente reformulado. O NS-3 foi construído como uma biblioteca, que pode ser conectada dinâmica ou estaticamente a um programa principal escrito em C++ que define a topologia de rede e inicializa o simulador. Além disso, o NS-3 também exporta praticamente todas as suas APIs para a linguagem Python, permitindo que *scripts* de simulação sejam escritos nesta linguagem.

A Figura 6.1 apresenta um diagrama com a organização dos módulos que compõem o NS-3. Para a implementação da plataforma de simulação, a versão do NS-3 utilizada foi a versão ns-3.12.

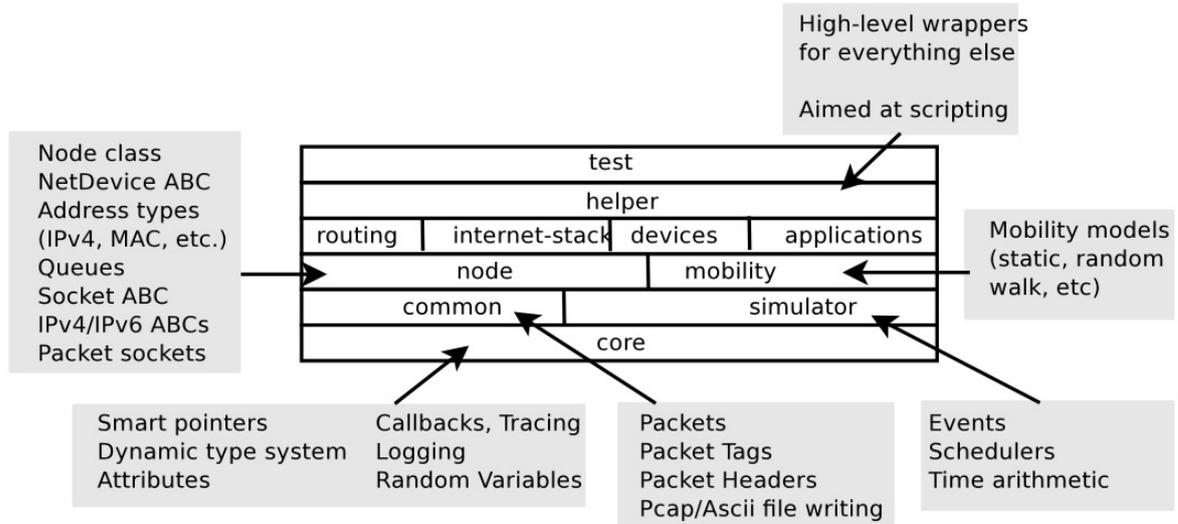


Figura 6.1: Organização do NS-3 [49].

6.2.2 Módulos Desenvolvidos

Apesar de ser um simulador de rede poderoso com boas características de escalabilidade, construído com modernas técnicas de programação e possibilitar integração com o ‘mundo real’ [50], o NS-3 é um ambiente relativamente novo e muitos módulos desenvolvidos para o NS-2 ainda não foram portados para o NS-3. Logo, é comum que pesquisadores que utilizem este ambiente hoje tenham que desenvolver uma parte dos módulos de que necessitam.

Para a plataforma de simulação proposta, foram desenvolvidos dois módulos de simulação, um módulo de configuração, além de alguns protocolos. A Figura 6.2 ilustra a topologia utilizada no simulador, a qual está em consonância com a topologia dos testes realizados em laboratório, descritos no Capítulo 4.

Na Figura 6.2 é possível identificar os dois módulos de simulação e o módulo de configuração desenvolvidos:

- *DnsQueryTool*: Este módulo é responsável por executar as funções realizadas pela ferramenta *dnstperf* no sistema real. A partir do número de clientes indicado pelo usuário, este módulo simula as requisições enviadas ao servidor de nomes autoritativo.
- *DnsAuthoritativeServer*: Este módulo contempla as funcionalidades de um servidor de nomes autoritativo baseado no software BIND. O comportamento deste módulo

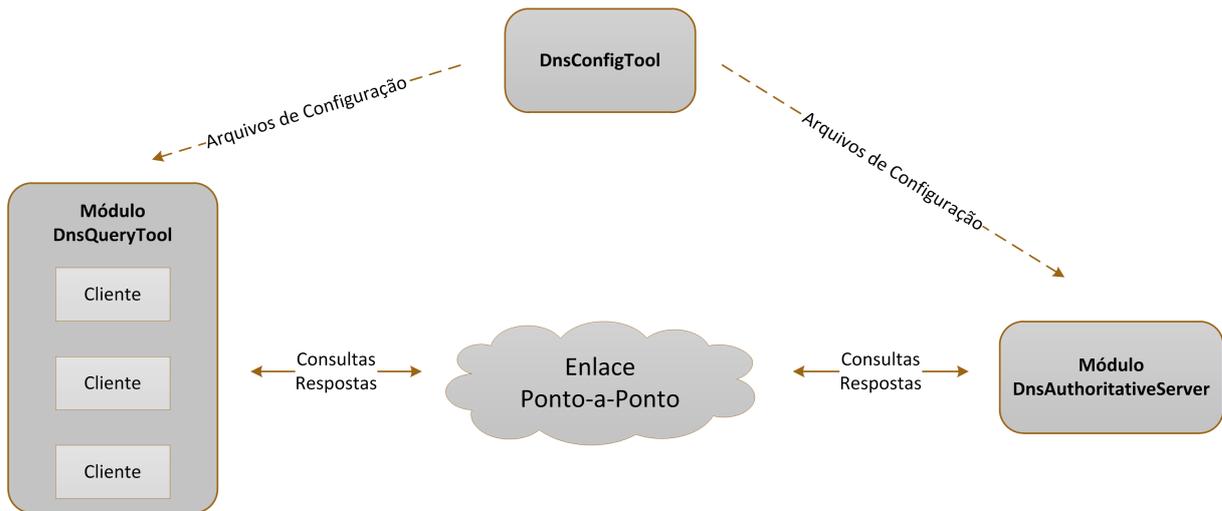


Figura 6.2: Topologia do Simulador.

é regido pelos modelos propostos no Capítulo 5.

- *DnsConfigTool*: Este módulo é responsável por gerar os arquivos de configuração requisitados pelos módulos de simulação.

6.2.3 Arquivos de Configuração

Como dito anteriormente, o projeto da plataforma de simulação foi desenvolvido de forma a ser o mais próximo da realidade. Sendo assim, os módulos de simulação requerem os mesmos arquivos de configuração tipicamente exigidos pelas entidades reais.

O uso de arquivos de configuração e o desenvolvimento do módulo *DnsConfigTool* têm por objetivo facilitar a utilização de futuros usuários da plataforma de simulação. O usuário precisa apenas modificar os arquivos de configuração sem precisar entender a complexidade do código que implementa os módulos.

O arquivo de configuração utilizado pelo módulo *DnsQueryTool* é um arquivo contendo as consultas a serem realizadas, o qual é exatamente idêntico ao arquivo de consultas requerido pela ferramenta *dnsperf*. Este arquivo é composto pelo FQDN para o qual deseja-se a resposta, seguido pelo tipo de registro de recurso desejado. Um trecho deste arquivo está ilustrado a seguir:

```
0.0.0.0.0.0.3.3.4.3.5.5.e164.arpa NAPTR
1.0.0.0.0.0.3.3.4.3.5.5.e164.arpa NAPTR
```

```
2.0.0.0.0.3.3.4.3.5.5.e164.arpa NAPTR
```

```
3.0.0.0.0.3.3.4.3.5.5.e164.arpa NAPTR
```

```
4.0.0.0.0.3.3.4.3.5.5.e164.arpa NAPTR
```

O software DNS/ENUM utilizado no teste em laboratório foi o software BIND. Logo, os arquivos de configuração para o módulo *DnsAuthoritativeServer* seguem exatamente a mesma sintaxe dos arquivos utilizados no software BIND. O arquivo *named.conf* contém as zonas sob a responsabilidade do servidor de nomes autoritativo. O conteúdo do arquivo *named.conf* está ilustrado abaixo:

```
zone "0.3.3.4.3.5.5.e164.arpa" {  
    type master;  
    file "db.0.3.3.4.3.5.5.e164.arpa";  
};
```

```
zone "1.3.3.4.3.5.5.e164.arpa" {  
    type master;  
    file "db.1.3.3.4.3.5.5.e164.arpa";  
};
```

Os arquivos para as zonas DNS (*zone files*) também possuem a mesma sintaxe dos arquivos reais. Um trecho deste arquivo está ilustrado a seguir:

```
$TTL 86400  
$ORIGIN 0.3.3.4.3.5.5.e164.arpa.  
@ IN SOA ns.0.3.3.4.3.5.5.e164.arpa. root.e164.arpa. (1 21600 3600 604800 3600)  
@ IN NS ns.0.3.3.4.3.5.5.e164.arpa.  
ns IN A 10.0.0.1  
  
0.0.0.0.0 NAPTR 0 0 "u" "sip+E2U" "!^.*$!sip:553433000000@enum.ufu.br!" .  
1.0.0.0.0 NAPTR 0 0 "u" "sip+E2U" "!^.*$!sip:553433000001@enum.ufu.br!" .  
2.0.0.0.0 NAPTR 0 0 "u" "sip+E2U" "!^.*$!sip:553433000002@enum.ufu.br!" .  
3.0.0.0.0 NAPTR 0 0 "u" "sip+E2U" "!^.*$!sip:553433000003@enum.ufu.br!" .  
4.0.0.0.0 NAPTR 0 0 "u" "sip+E2U" "!^.*$!sip:553433000004@enum.ufu.br!" .
```

Maiores detalhes sobre a sintaxe dos arquivos de configuração utilizados pela ferramenta *dnsperf* e pelo software BIND, podem ser encontrados em [44] e [51], respectivamente.

Estes arquivos de configuração não são uma preocupação para um usuário típico da plataforma de simulação, uma vez que o módulo *DnsConfigTool* é responsável por criar este arquivos e movê-los para os diretórios adequados para a correta utilização por parte dos módulos de simulação.

Para o usuário é necessário apenas fornecer um arquivo de configuração simples com as opções desejadas. Este arquivo guiará o módulo *DnsConfigTool* na criação dos arquivos de configuração de acordo com as opções selecionadas pelo usuário. Este arquivo de opções está ilustrado a seguir:

```
// Setup configuration file
numOfRecords      500k
numOfZones        10
domainName        e164.arpa
serverAddress     10.0.0.1
```

Futuramente, este arquivo poderá ser substituído por uma interface gráfica para melhorar a experiência de usabilidade do simulador. A Figura 6.3 sumariza os arquivos de configuração utilizados na plataforma de simulação.

6.3 Implementação do Módulo *DnsQueryTool*

O módulo *DnsQueryTool* é uma classe C++ que herda a classe base *Application* pertencente ao NS-3. A Figura 6.4 traz um diagrama com as principais ações deste módulo durante a simulação.

Quando a simulação é iniciada, o *script* de simulação informa ao módulo *DnsQueryTool* o número de clientes que serão simulados. Este número determina, então, o número máximo de consultas que poderão estar em andamento, isto é, aguardando a resposta vinda do servidor de nomes autoritativo.

Antes de enviar qualquer requisição, o módulo determina a quantidade de consultas em andamento e verifica se este número é menor que o número máximo de consultas permitido. Caso este número seja maior, o módulo entra em espera e agenda um atraso

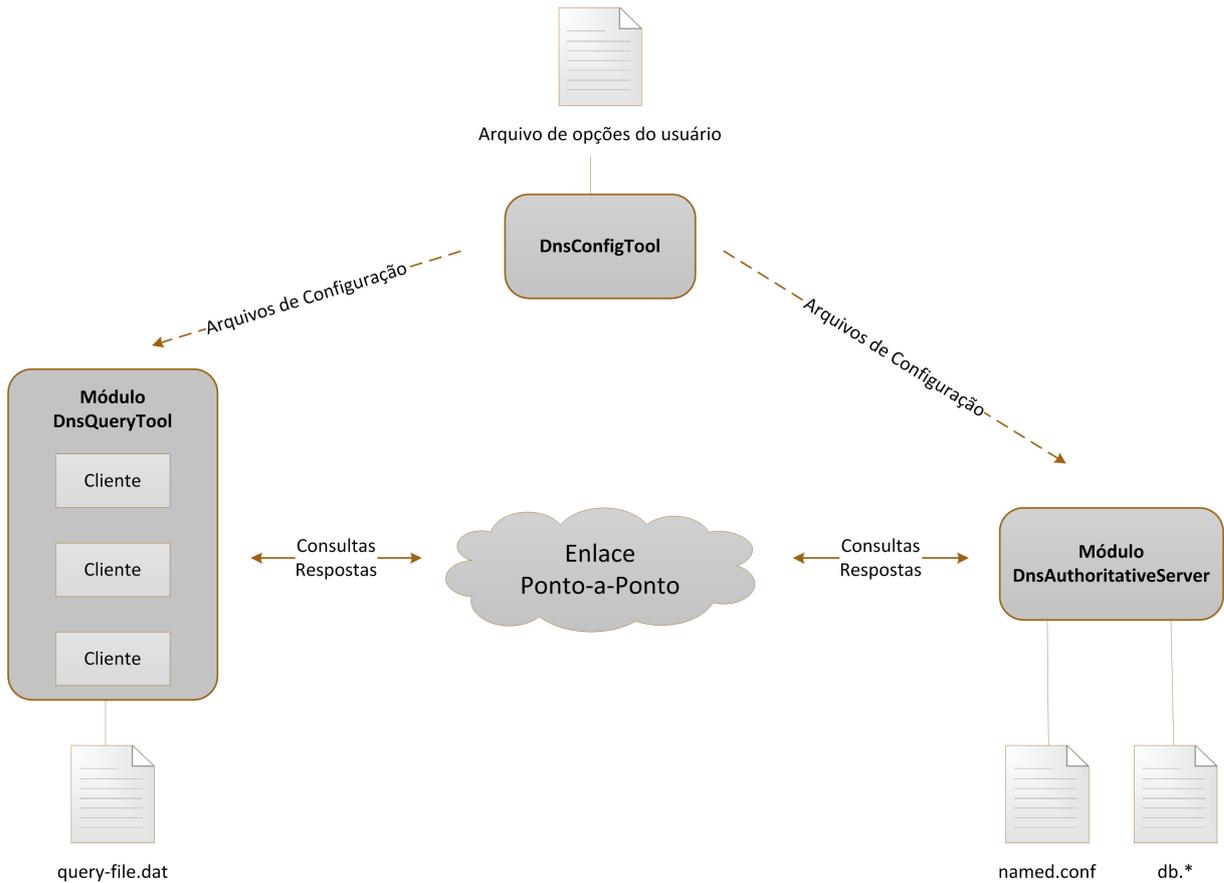


Figura 6.3: Arquivos de configuração requeridos por cada módulo da plataforma de simulação.

de regulação do tráfego (*throttle*). A ferramenta *dnstperf* estabelece que este atraso deve ser de 25 ms. Após esperar este intervalo, o módulo realiza nova verificação do número de consultas em andamento para saber se pode proceder o envio da requisição.

Uma vez que a requisição pode ser enviada, o módulo procura no arquivo de consultas qual deve ser o FQDN consultado e o tipo de registro de recurso desejado. Após enviar a requisição, o módulo agenda um *timeout* que é utilizado para considerar uma consulta perdida. Após um intervalo de 5 s, se a resposta para a consulta em questão não for recebida o módulo contabiliza uma perda, decrementa o número de consultas em andamento e verifica a possibilidade de envio de uma nova consulta. Após o envio da consulta, o módulo deve também incrementar o número de consultas em andamento.

Quando recebe um pacote de resposta do servidor autoritativo, o módulo calcula o tempo necessário para receber a resposta para aquela requisição. O próximo passo é decrementar o número de consultas em andamento e verificar se o envio de uma nova

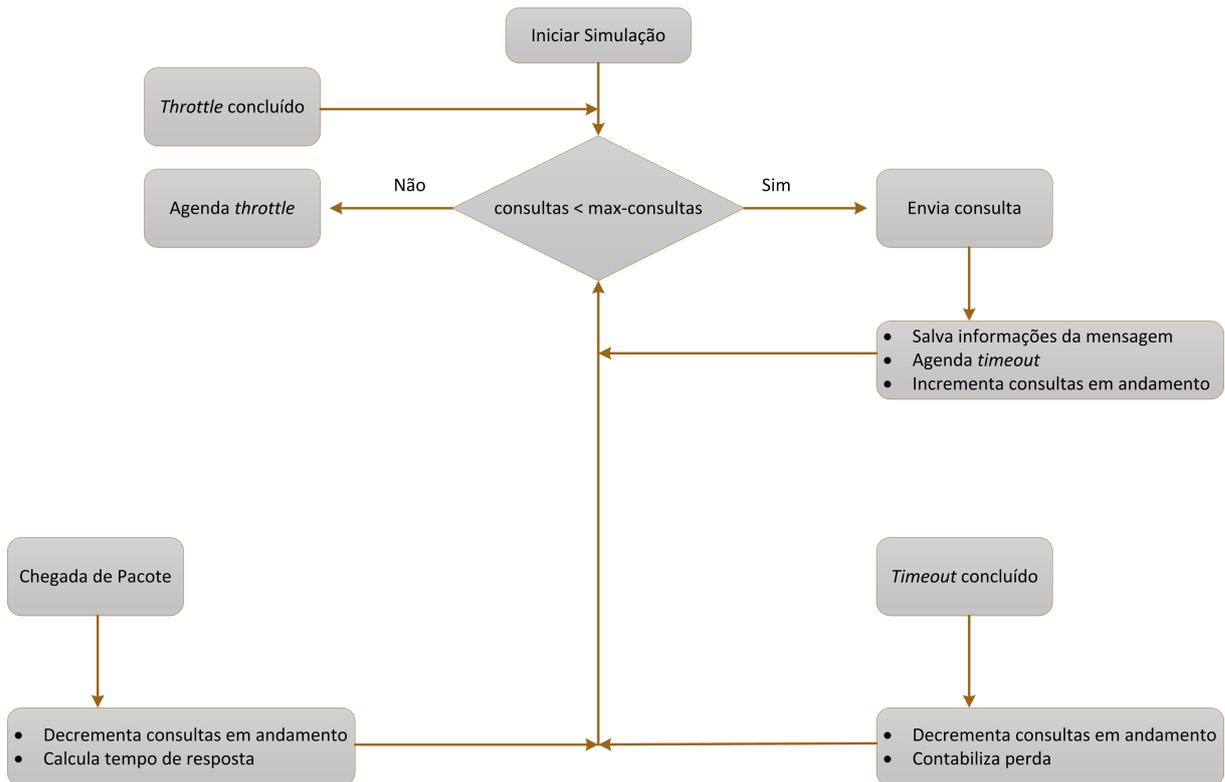


Figura 6.4: Diagrama de fluxo para o módulo *DnsQueryTool*.

consulta está liberado.

6.4 Implementação do Módulo *DnsAuthoritativeServer*

O módulo *DnsAuthoritativeServer* é uma classe C++ que herda a classe base *Application* pertencente ao NS-3. A Figura 6.5 traz um diagrama com as principais ações deste módulo durante a simulação.

Quando a simulação é iniciada, o módulo *DnsAuthoritativeServer* inicia o processamento do arquivo *named.conf* para determinar quais são as zonas pelas quais é responsável. O arquivo *named.conf* também informa a localização dos arquivos de zonas com os registros de recurso. Em um segundo instante o módulo processa os arquivos *db.** que possuem os registros NAPTR. Após concluída esta etapa de carregamento dos arquivos na memória, o módulo entra no modo de espera por requisições enviadas pelos clientes.

Quando um pacote de consulta é recebido, o módulo *DnsAuthoritativeServer* faz o cálculo do atraso de processamento (serviço) daquele pacote, ou seja, qual o tempo gasto para encontrar a resposta para aquela consulta. Este cálculo é baseado na modelagem

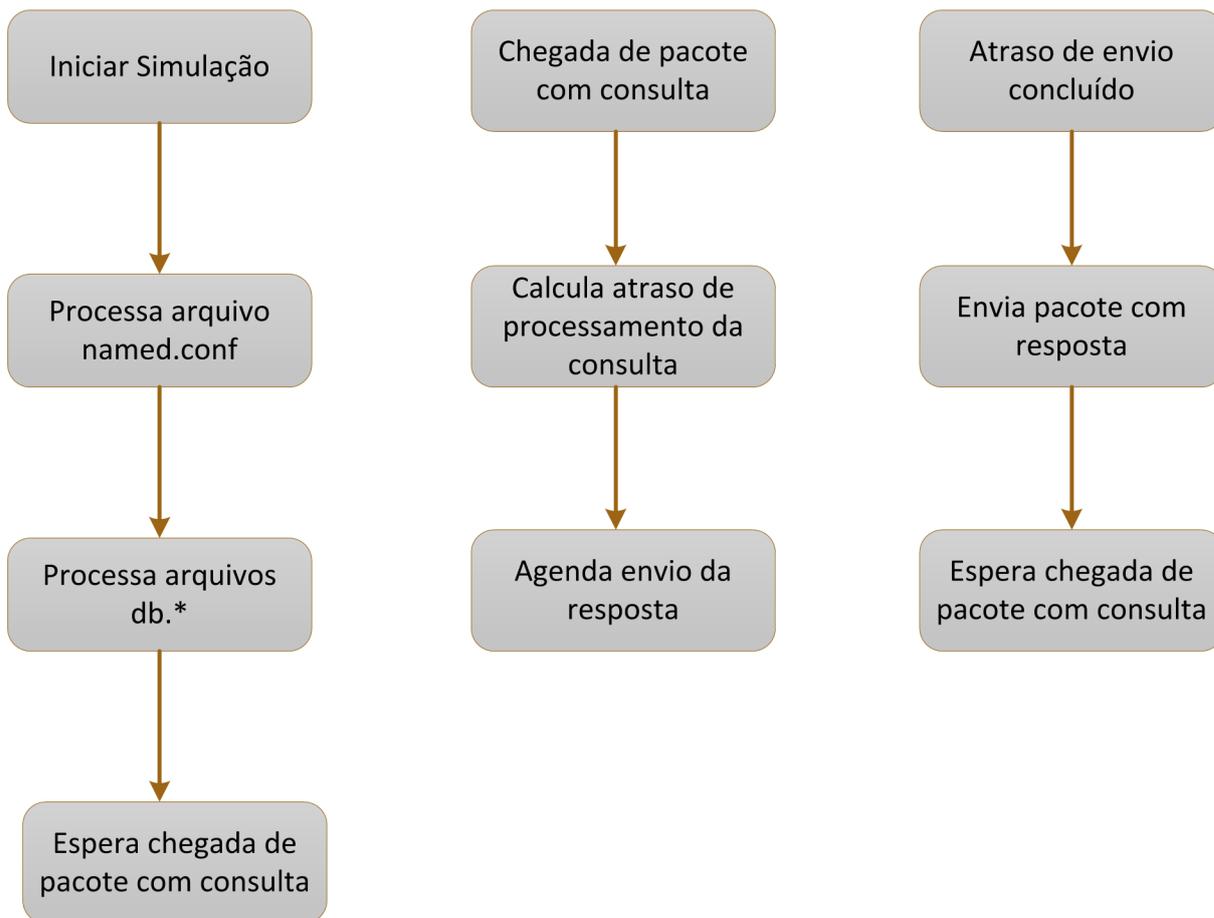


Figura 6.5: Diagrama de fluxo para o módulo *DnsAuthoritativeServer*.

realizada para o servidor de nomes real, que foi descrita no Capítulo 5. A partir dos dados de tempo médio de resposta obtidos nos testes em laboratório e do desvio padrão deste tempo médio, o módulo calcula o tempo médio de serviço daquela requisição utilizando uma distribuição normal. Após realizar este cálculo, o módulo *DnsAuthoritativeServer* agenda o envio da resposta de acordo com o tempo médio de serviço determinado anteriormente.

Quando o instante agendado no passo anterior é alcançado, o módulo *DnsAuthoritativeServer* envia a resposta para o módulo *DnsQueryTool* e entra novamente no modo de espera pela chegada de pacotes de consulta.

6.5 Integração do Simulador ao NS-3

Como dito anteriormente, o ambiente de simulação escolhido para desenvolver a plataforma de simulação foi o NS-3. Esta seção descreve as ações realizadas para integrar o código

da plataforma de simulação ao NS-3.

Além dos módulos *DnsQueryTool* e *DnsAuthoritativeServer*, foi necessário implementar outras classes para dar suporte ao funcionamento destes módulos. Todas estas novas classes implementadas (assim como os módulos *DnsQueryTool* e *DnsAuthoritativeServer*) foram escritas em C++ de forma que tivessem total compatibilidade com o código do NS-3. A Figura 6.6 ilustra os arquivos com as classes que foram adicionadas à estrutura do NS-3 para a implementação da plataforma de simulação. As próximas subseções descrevem estas classes utilizadas.

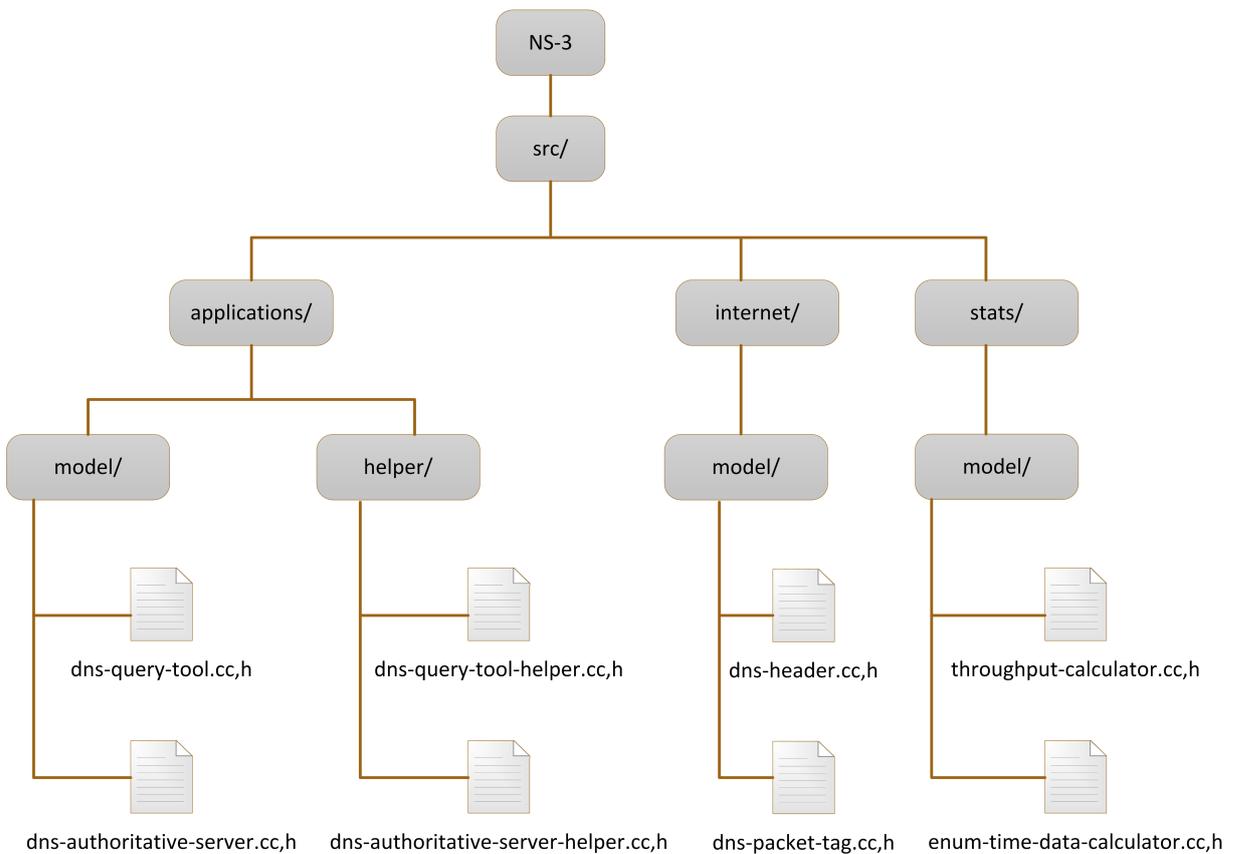


Figura 6.6: Integração da plataforma de simulação ao NS-3.

6.5.1 Classe *DnsHeader*

A classe *DnsHeader* é descrita nos arquivos *dns-header.h* e *dns-header.cc*. Esta classe herda a classe base *Header* pertencente ao NS-3. O propósito desta classe é implementar o cabeçalho e o conteúdo de uma mensagem DNS. O NS-3 possui uma filosofia de modelar os cabeçalhos reais da forma mais fiel possível. Seguindo esta filosofia, a implementação

do cabeçalho DNS foi realizada de forma que este cabeçalho fosse exatamente o mesmo que é transmitido em um sistema real, sem qualquer abstração.

Para confirmar esta hipótese utilizou-se um capturador de pacotes para comparar o cabeçalho de uma consulta real com o cabeçalho de uma consulta em ambiente simulado. Felizmente, o NS-3 oferece a possibilidade de exportar os pacotes que transitaram pela rede durante a simulação para o formato *.pcap*. Utilizando o software Wireshark [52], um dos mais populares analisadores de pacotes de rede, realizou-se a comparação dos cabeçalhos. As Figuras 6.7 e 6.8 mostram o resultado da captura de dois pacotes contendo mensagens de consulta DNS distintas, em ambientes real e simulado, respectivamente.

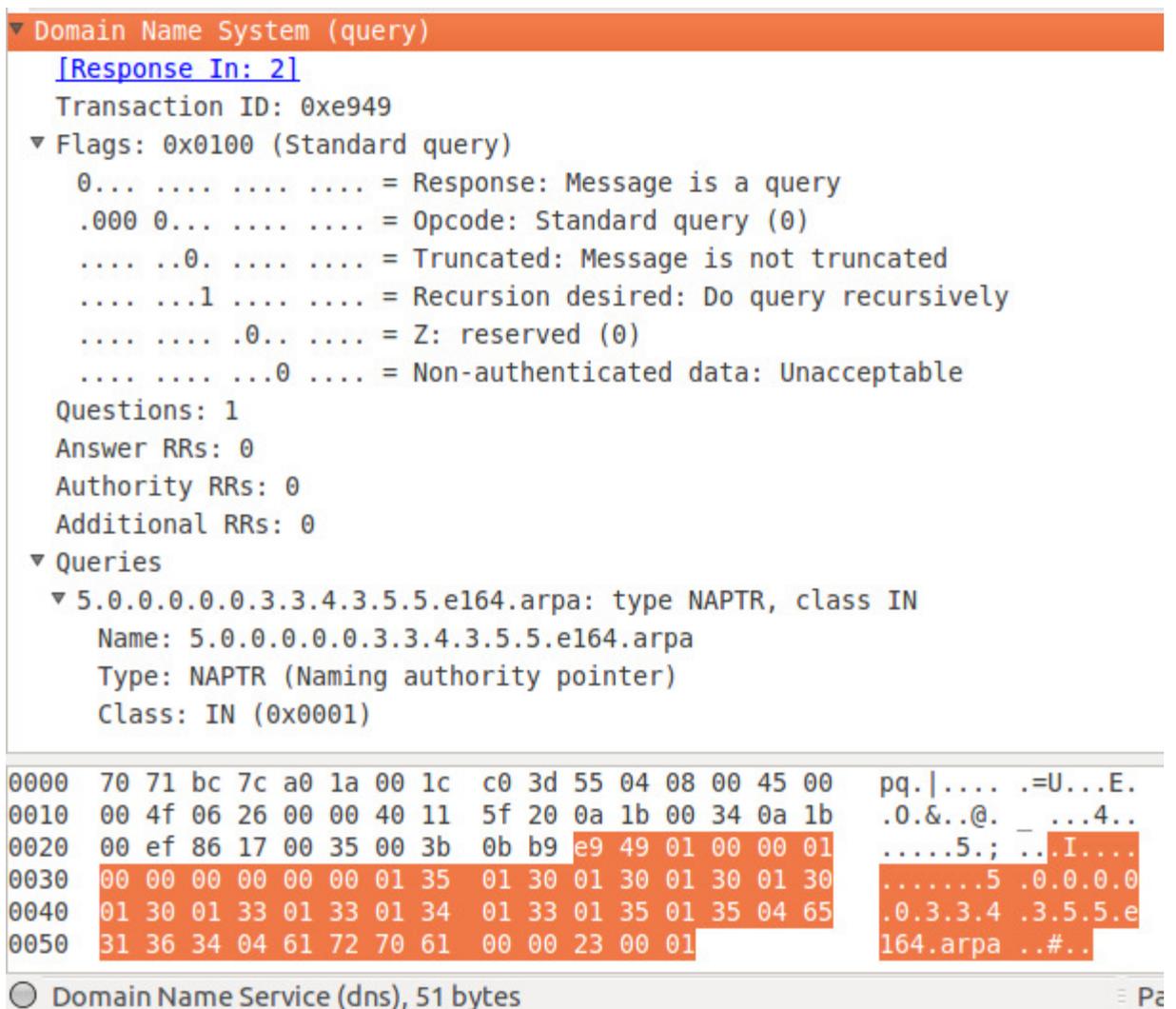


Figura 6.7: Cabeçalho DNS no sistema real.

Nas figuras é possível identificar que em ambos pacotes a mensagem DNS contém 51 bytes de dados. O cabeçalho DNS só difere no identificador da mensagem, o que já era

```

▼ Domain Name System (query)
  Transaction ID: 0xffff
  ▼ Flags: 0x0100 (Standard query)
    0... .. = Response: Message is a query
    .000 0... .. = Opcode: Standard query (0)
    .... ..0. .... = Truncated: Message is not truncated
    .... ..1 .... = Recursion desired: Do query recursively
    .... ..0.. .... = Z: reserved (0)
    .... ..0 .... = Non-authenticated data: Unacceptable
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  ▼ Queries
    ▼ 0.0.0.0.0.0.3.3.4.3.5.5.e164.arpa: type NAPTR, class IN
      Name: 0.0.0.0.0.0.3.3.4.3.5.5.e164.arpa
      Type: NAPTR (Naming authority pointer)
      Class: IN (0x0001)
  
```

0000	00 21 45 00 00 4f 00 00 00 00 40 11 00 00 0a 00	!.E..0.. ..@.....
0010	00 02 0a 00 00 01 c0 01 00 35 00 3b 00 00 ff ff5.;....
0020	01 00 00 01 00 00 00 00 00 00 01 30 01 30 01 300.0.0
0030	01 30 01 30 01 30 01 33 01 33 01 34 01 33 01 35	.0.0.0.3 .3.4.3.5
0040	01 35 04 65 31 36 34 04 61 72 70 61 00 00 23 00	.5.e164. arpa..#.
0050	01	.

● Domain Name Service (dns), 51 bytes

Figura 6.8: Cabeçalho DNS no ambiente de simulação.

esperado, uma vez que este identificador é atribuído aleatoriamente. Da mesma forma, o conteúdo da mensagem de consulta é o mesmo para o pacote real e o pacote obtido via simulação, com exceção, é claro, do FQDN consultado. A seção na parte inferior das figuras, com a visualização hexadecimal dos bits transmitidos (destacado na cor laranja), finaliza a validação do cabeçalho DNS implementado para a plataforma de simulação. A mensagem de resposta DNS é mais complexa que a mensagem de consulta, mas recebeu o mesmo rigor na implementação e foi validada de forma análoga à mensagem de consulta.

6.5.2 Classe *DnsPacketTag*

A classe *DnsPacketTag* é descrita nos arquivos *dns-packet-tag.h* e *dns-packet-tag.cc*. Esta classe herda a classe base *Tag* pertencente ao NS-3. O propósito da classe *DnsPacket-*

Tag é adicionar uma marca de tempo que registra o instante em que o pacote foi enviado pelo módulo *DnsQueryTool*. Ao receber o pacote de volta, o módulo *DnsQueryTool* é capaz de calcular o tempo de resposta para este pacote.

6.5.3 Classe *EnumTimeDataCalculator*

A classe *EnumTimeDataCalculator* é descrita nos arquivos *enum-time-data-calculator.h* e *enum-time-data-calculator.cc*. Esta classe herda a classe base *MinMaxAvgTotalCalculator* pertencente ao NS-3. A classe *EnumTimeDataCalculator* é utilizada para fins estatísticos, sendo responsável pelo cálculo do tempo médio de resposta e do desvio padrão do tempo médio de resposta.

6.5.4 Classe *ThroughputCalculator*

A classe *ThroughputCalculator* é descrita nos arquivos *throughput-calculator.h* e *throughput-calculator.cc*. Esta classe herda a classe base *DataCalculator* que pertence ao NS-3. A classe *ThroughputCalculator* é responsável por calcular a vazão de resposta às consultas do servidor de nomes autoritativo.

6.5.5 Classe *DnsQueryToolHelper*

A classe *DnsQueryToolHelper* é descrita nos arquivos *dns-query-tool-helper.h* e *dns-query-tool-helper.cc*. A classe *DnsQueryToolHelper* é uma classe auxiliar que torna mais fácil instanciar a classe *DnsQueryTool* ao escrever um *script* de simulação.

6.5.6 Classe *DnsAuthoritativeServerHelper*

A classe *DnsAuthoritativeServerHelper* é descrita nos arquivos *dns-authoritative-server-helper.h* e *dns-authoritative-server-helper.cc*. A classe *DnsAuthoritativeServerHelper* é uma classe auxiliar que torna mais fácil instanciar a classe *DnsAuthoritativeServer* ao escrever um *script* de simulação.

6.6 Módulo *DnsConfigTool*

Na Seção 6.2.2 foram apresentados três módulos: dois módulos de simulação (*DnsQueryTool* e *DnsAuthoritativeServer*) e um módulo de configuração (*DnsConfigTool*). Diferentemente dos módulos de simulação, o módulo *DnsConfigTool* não foi escrito em C++, mas em Python, uma vez que o Python é uma poderosa linguagem de *script* e não existem restrições quanto a velocidade de execução do programa.

O módulo *DnsConfigTool* foi construído de forma independente em relação ao ambiente de simulação NS-3 já que seu propósito é apenas criar e gerenciar os arquivos de configuração apresentados na Seção 6.2.3. Este módulo é composto por algumas classes com diferentes funções que são descritas nas subseções a seguir.

6.6.1 Classes *FileParser*, *UnityParser* e *SetupTool*

Estas três classes são responsáveis por processar o arquivo de opções informado pelo usuário com os parâmetros de simulação. Estas são classes auxiliares cujos métodos são compartilhadas pelas classes *QueryFileGenerator* e *ZoneFilesGenerator*.

6.6.2 Classes *QueryFileGenerator* e *ZoneFilesGenerator*

As classes *QueryFileGenerator* e *ZoneFilesGenerator* são responsáveis pela geração dos arquivos de configuração para os módulos *DnsQueryTool* e *DnsAuthoritativeServer*, respectivamente. A partir do arquivo com as opções do usuário, estas classes criam os arquivos de configuração necessários e copiam estes arquivos para os diretórios adequados, de forma transparente para o usuário.

6.7 Validação da Plataforma de Simulação

Uma vez que todos os módulos de simulação estavam concluídos, procedeu-se a realização de testes simulados para verificar a exatidão do modelo de simulação.

O *script* de teste considerou os valores estabelecidos para o teste real em laboratório:

- O enlace de interconexão tem largura de banda de 1 Gbps e atraso de propagação de 25 ns;

- O tempo de simulação foi estabelecido em 60 segundos;
- O tamanho do banco de dados foi definido como sendo de 500k registros.

Assim como nos testes em laboratório, os testes simulados foram divididos em etapas. A cada etapa, o número de clientes simulado foi incrementado. Além disso, cada etapa de teste foi realizada cinco vezes com diferentes valores de *seed* para assegurar a aleatoriedade dos resultados. O resultado final é a média aritmética destas cinco amostras.

Como descrito no Capítulo 5, este trabalho almeja analisar duas estratégias de modelagem para o servidor de nomes autoritativo e verificar qual delas oferece resultados mais próximos daqueles encontrados nos testes em laboratório.

A primeira abordagem de modelo para o servidor autoritativo é baseada na regressão de funções. Considerando as equações apresentadas na Tabela 5.1, o simulador calcula o intervalo de serviço para cada requisição que recebe. As Figuras 6.9, 6.10 e 6.11 mostram os resultados encontrados.

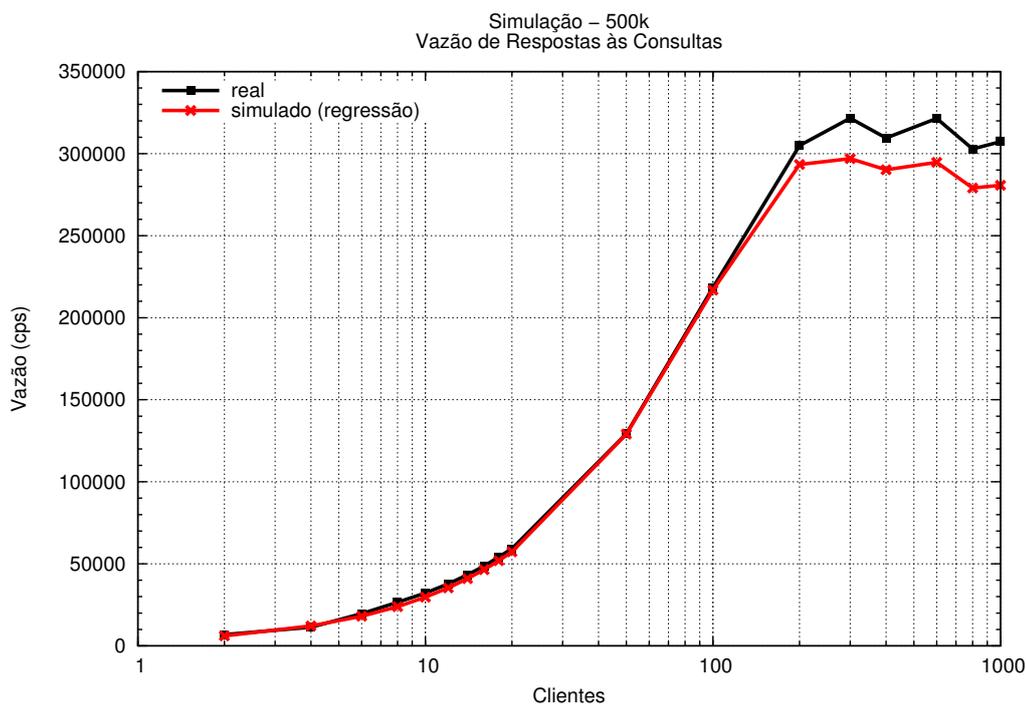


Figura 6.9: Desempenho do servidor DNS/ENUM baseado em simulação com relação a vazão de respostas às consultas (modelado por regressão de funções).

A segunda abordagem de modelo para o servidor autoritativo é baseada na utilização direta das amostras obtidas nos testes em laboratório. Considerando os dados apresenta-

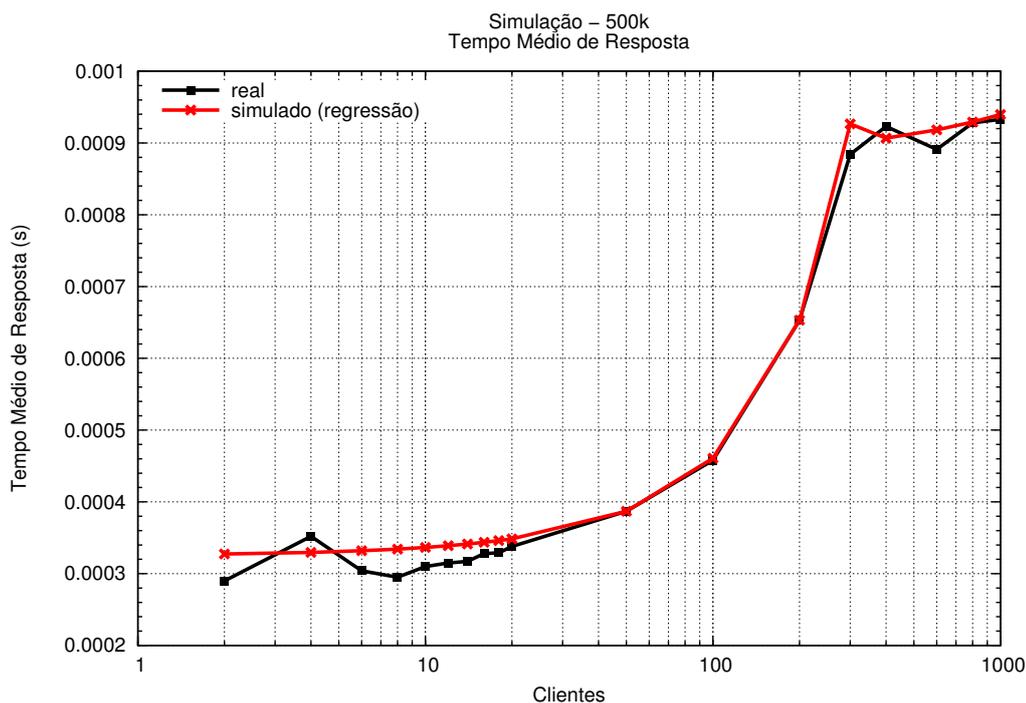


Figura 6.10: Comportamento do tempo médio de resposta baseado em simulação (modelado por regressão de funções).

dos na Tabela A.1, o simulador determina o intervalo de serviço para cada requisição que recebe. As Figuras 6.13, 6.14 e 6.15 mostram os resultados encontrados.

Ao realizar uma análise gráfica dos resultados apresentados nas figuras, observa-se que ambas estratégias obtiveram um bom desempenho na tentativa de modelar o comportamento real do servidor de nomes autoritativo. Entretanto, é necessário verificar qual abordagem ofereceu resultados mais próximos dos resultados reais. Neste sentido, uma comparação foi realizada com relação ao erro causado por cada estratégia. O cálculo do erro levou em consideração a diferença entre os resultados obtidos nos testes em laboratório com os resultados obtidos via simulação utilizando os dois modelos. O resultado dessa comparação está ilustrado nas Figuras 6.17, 6.18 e 6.19.

Analisando as figuras, é possível observar que a modelagem baseada no uso direto das amostras oferece resultados mais próximos dos resultados reais quando comparada à modelagem baseada em regressão de funções. Isto já era esperado, uma vez que a modelagem por amostras diminui as fontes de erro, por utilizar o valor exato obtido no sistema real, possibilitando assim resultados mais confiáveis.

Os resultados apresentados nas Figuras 6.17, 6.18 e 6.19 podem ser tomados como

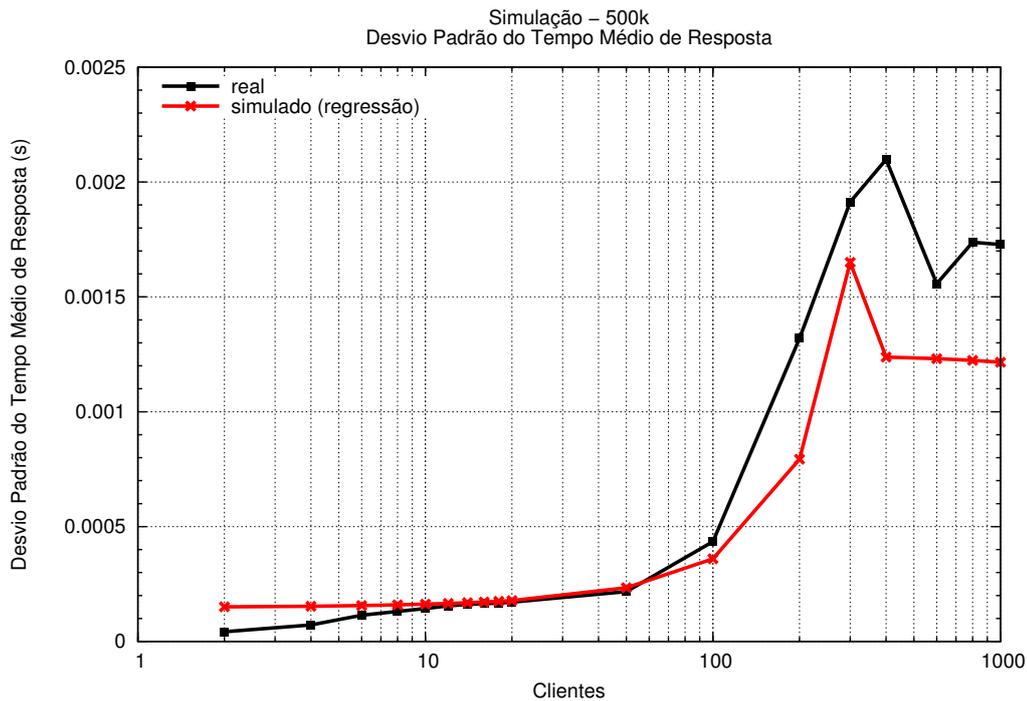


Figura 6.11: Comportamento do desvio padrão do tempo médio de resposta baseado em simulação (modelado por regressão de funções).

comprovação do sucesso do simulador e utilizados na validação da proposta desta dissertação, uma vez que o erro encontrado para a modelagem baseada na utilização de amostras ficou dentro de patamares satisfatórios.

Vale ressaltar ainda que, embora a modelagem baseada na utilização direta das amostras apresentou um melhor desempenho, esta estratégia possui algumas limitações quanto as possibilidades de cenários disponíveis. Dessa forma, a plataforma de simulação adota as duas estratégias de acordo com o cenário em estudo. O simulador utiliza preferencialmente o modelo por amostras. Para os cenários que não foram realizados em laboratório, cujos os dados não estão disponíveis, o simulador utiliza o modelo baseado na regressão de funções.

Por fim, a Figura 6.20 apresenta o intervalo de confiança para a vazão de respostas às consultas, considerando-se o modelo baseado na utilização direta de amostras. Nesta figura é possível notar que a faixa do intervalo de confiança é tão estreita que só é parcialmente visível na zona de saturação do servidor, onde o simulador tem uma menor acurácia. Para visualizar melhor esta faixa, a Figura 6.21 exhibe os dados com um *zoom* na zona de saturação. Este comportamento do intervalo de confiança foi observado nas

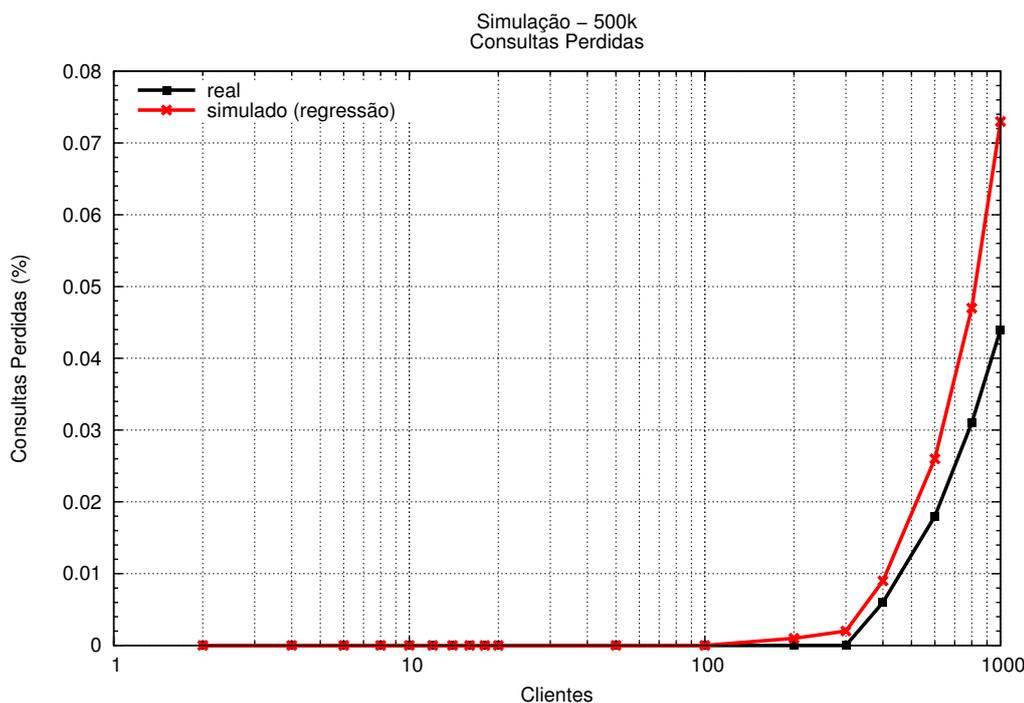


Figura 6.12: Comportamento da taxa de perdas baseada em simulação (modelado por regressão de funções).

demais métricas avaliadas, e para ambas as estratégias de modelagem.

6.8 Considerações Finais

Neste capítulo foi apresentado todo o processo de implementação, integração e validação do simulador proposto. A implementação do modelo de simulação foi realizada tendo em vista o design do ambiente de simulação NS-3, de forma que a integração dos módulos construídos ocorresse de forma adequada.

Como o NS-3 é um simulador relativamente novo, foi necessário desenvolver classes auxiliares para suprir algumas lacunas existentes neste ambiente de simulação. O exemplo mais notável foi a implementação fidedigna das mensagens DNS de consulta e resposta.

O capítulo descreveu ainda o módulo de configuração *DnsConfigTool*, idealizado para ser uma interface com o usuário, facilitando a utilização da plataforma de simulação.

Por fim, foi possível observar que os resultados de simulação foram obtidos satisfatoriamente. As curvas apresentadas comprovaram que o modelo baseado na utilização direta das amostras obtidas nos testes em laboratório oferece resultados plausíveis e confiáveis.

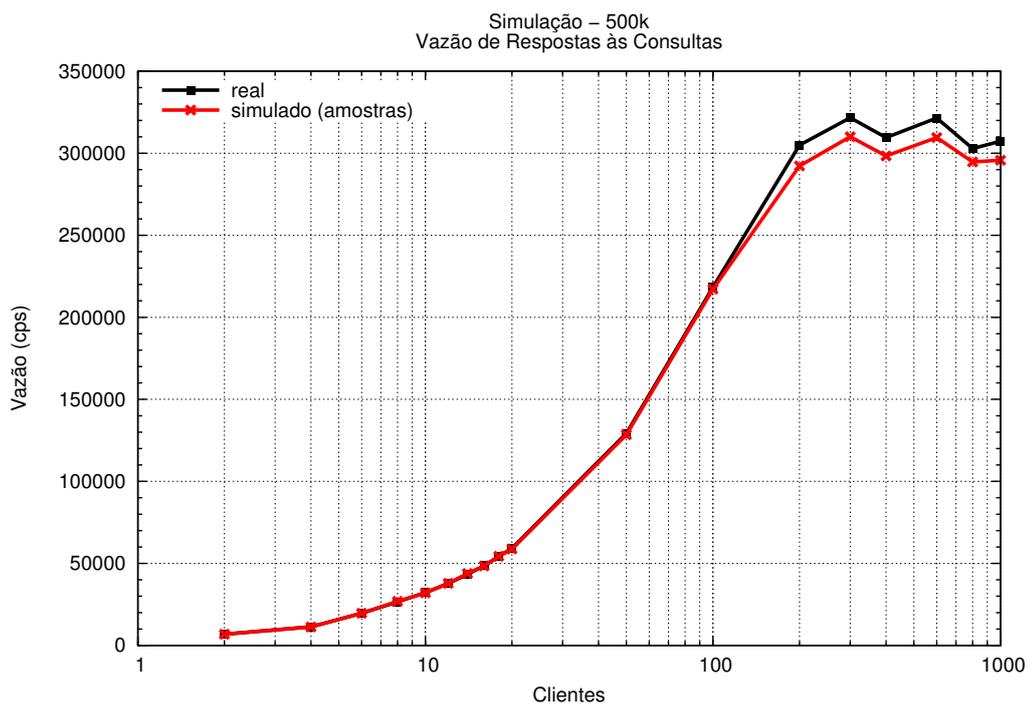


Figura 6.13: Desempenho do servidor DNS/ENUM baseado em simulação com relação a vazão de respostas às consultas (modelado por amostras).

Além disso, foi possível comprovar também que este modelo supera o desempenho do modelo baseado na regressão de funções, conforme havia sido previsto no Capítulo 5.

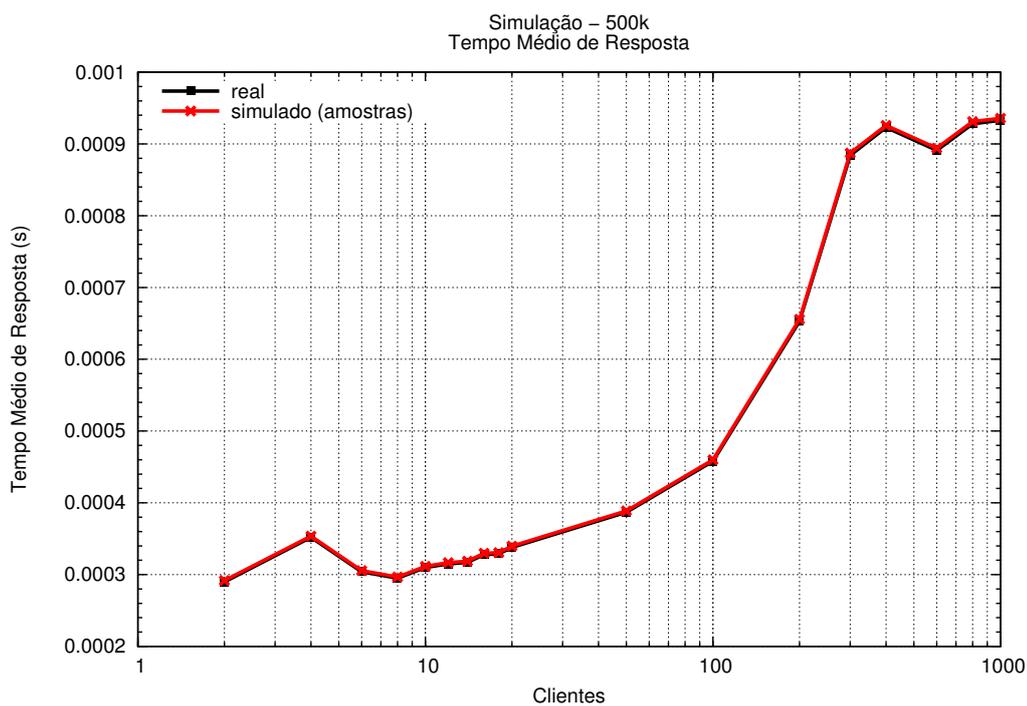


Figura 6.14: Comportamento do tempo médio de resposta baseado em simulação (modelado por amostras).

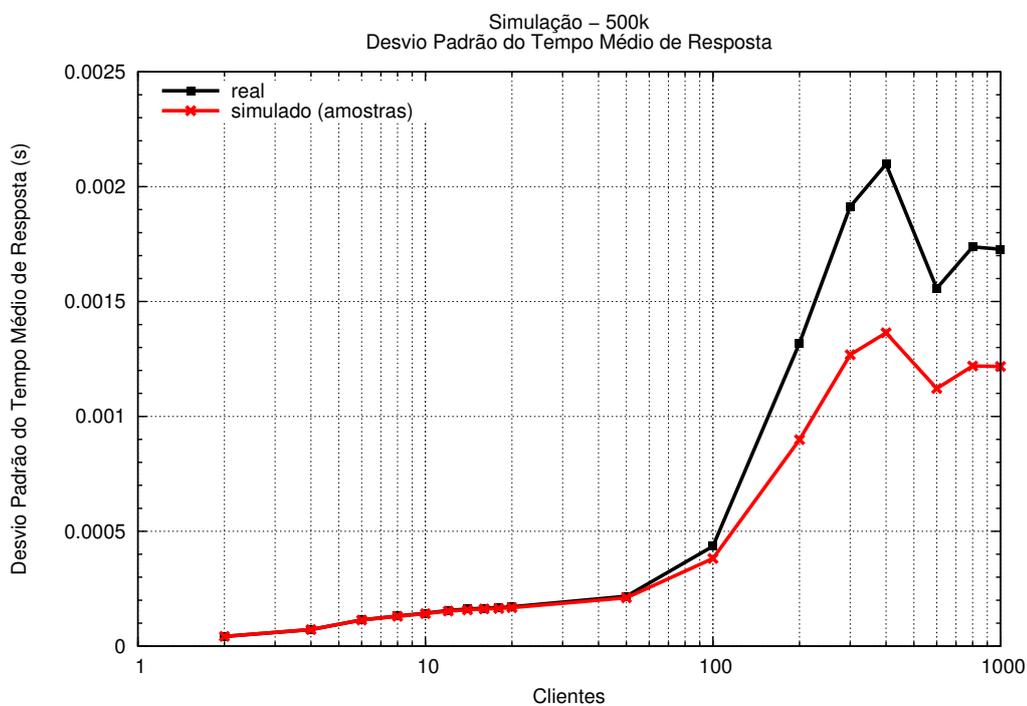


Figura 6.15: Comportamento do desvio padrão do tempo médio de resposta baseado em simulação (modelado por amostras).

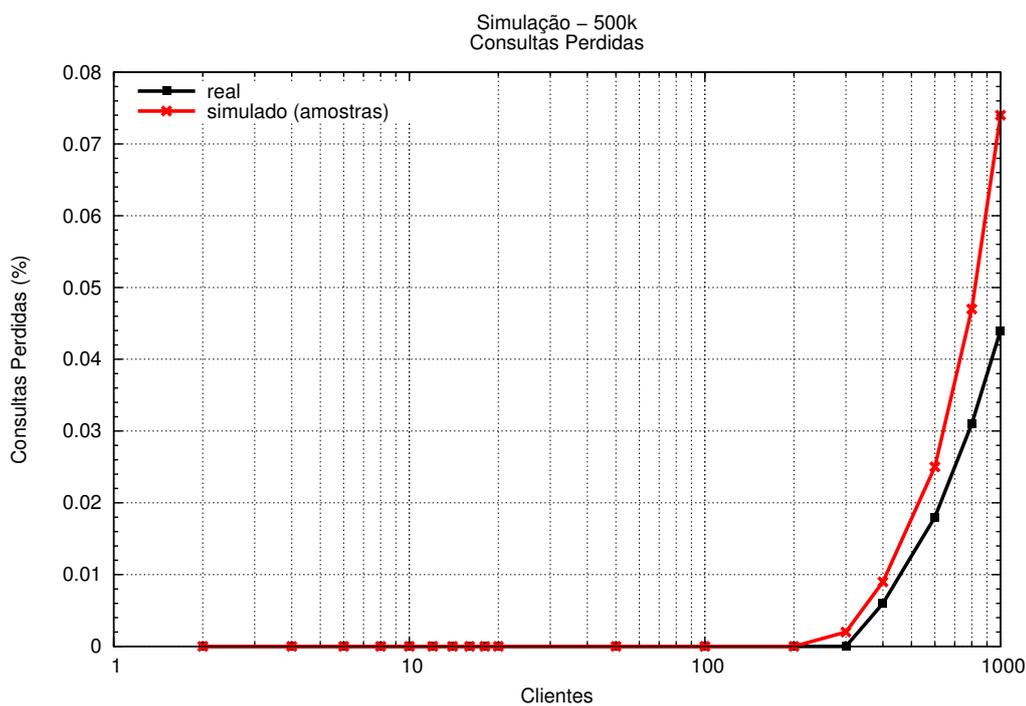


Figura 6.16: Comportamento da taxa de perdas baseado em simulação (modelado por amostras).

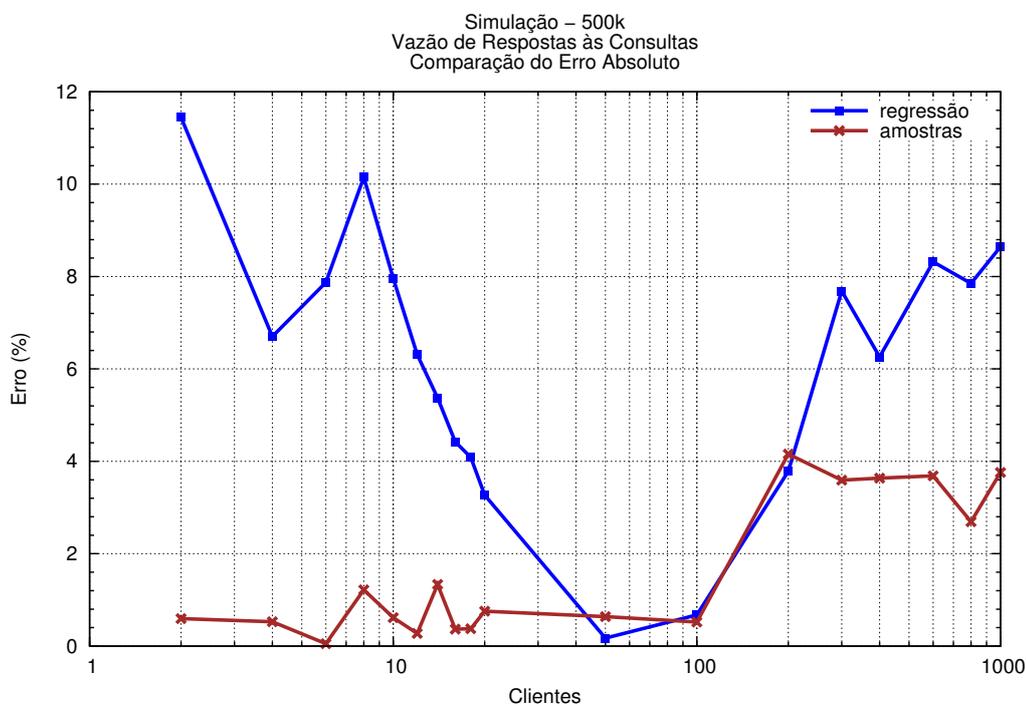


Figura 6.17: Comparação do erro cometido por cada estratégia de modelagem com relação a vazão de respostas às consultas.

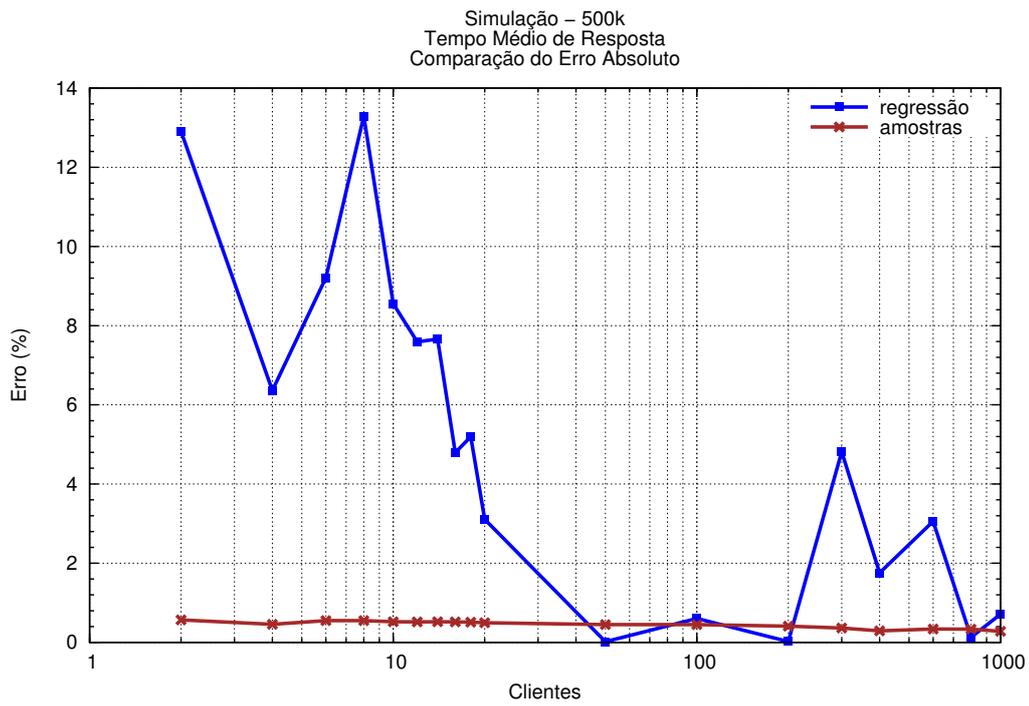


Figura 6.18: Comparação do erro cometido por cada estratégia de modelagem com relação ao tempo médio de resposta.

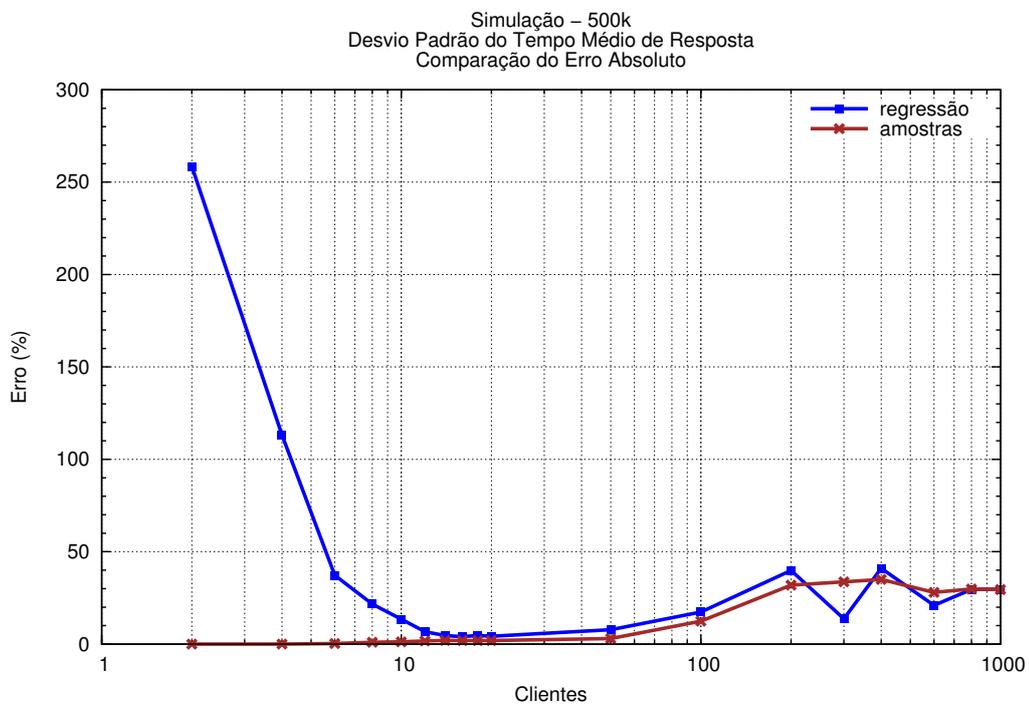


Figura 6.19: Comparação do erro cometido por cada estratégia de modelagem com relação ao desvio padrão do tempo médio de resposta.

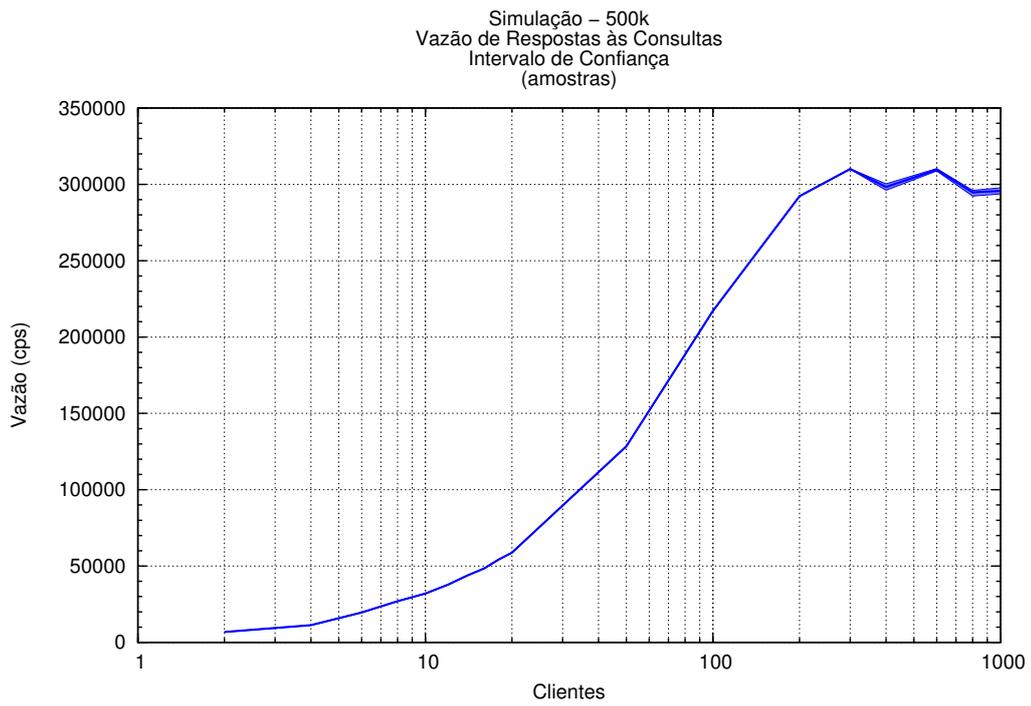


Figura 6.20: Intervalo de confiança (95%) para a vazão de respostas às consultas, considerando-se o modelo baseado na utilização direta de amostras.

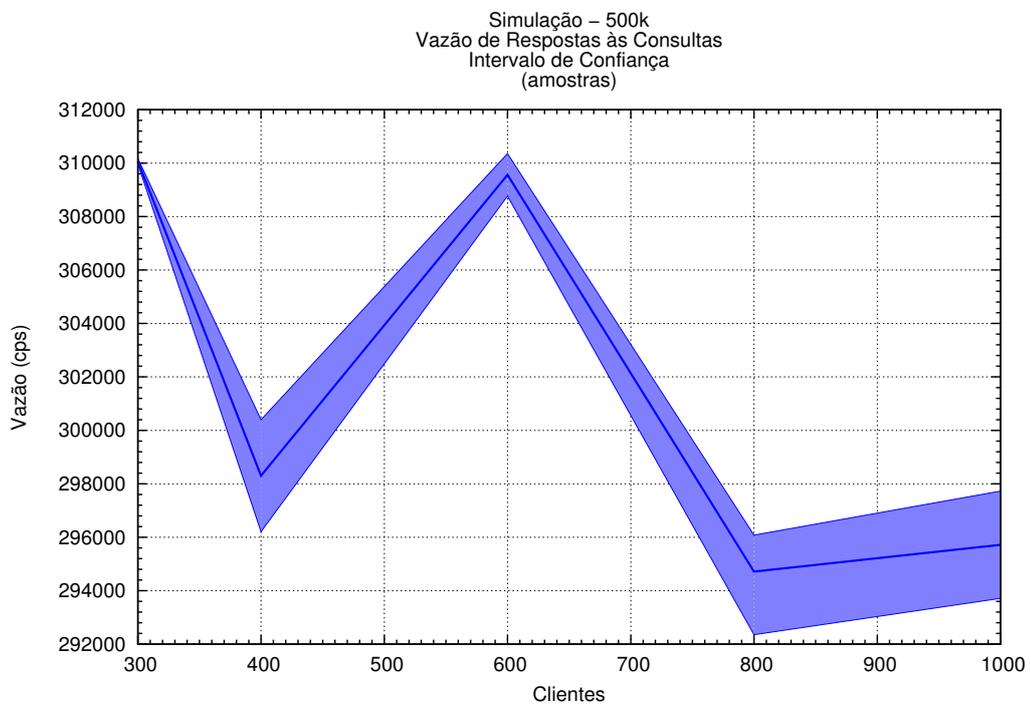


Figura 6.21: Intervalo de confiança (95%) para a vazão de respostas às consultas, considerando-se o modelo baseado na utilização direta de amostras.

Capítulo 7

Conclusões Gerais

Devido à crescente necessidade de integração entre as diferentes redes de comunicações existentes, o protocolo ENUM vem ganhando destaque como a solução mais promissora. O ENUM vem se posicionando como uma tecnologia madura e integradora que permitirá que operadoras e usuários usufruam de chamadas telefônicas com custos mais baixos que os atuais. Entretanto, a sua disseminação ainda é baixa, uma vez que a implantação está ocorrendo de forma gradual e de acordo com o interesse de cada país. Além disso, existem questões técnicas e, principalmente, interesses políticos e econômicos que precisam ser trabalhados para que este protocolo seja popularizado.

Diante dos desafios que o ENUM apresenta para ser implantado com sucesso, existe uma demanda por estudos que permitam o conhecimento do comportamento deste protocolo. Dentre as estratégias para a realização deste estudo, a modelagem e simulação se posiciona como uma opção barata, flexível, que proporciona resultados próximos aos de um situação real. Considerando-se que não existem muitos trabalhos nessa área e sabendo-se ainda que existe uma carência de ferramentas para estudo por simulação, esta dissertação propôs uma plataforma de simulação para a realização de estudos do comportamento do protocolo ENUM.

Partindo do pressuposto de que este simulador deveria oferecer resultados confiáveis, o projeto de concepção estabeleceu um sistema real simples como objeto de estudo. Por outro lado, este sistema simples pôde ser reproduzido em laboratório, possibilitando a obtenção de resultados reais para a posterior validação do simulador proposto.

Os testes realizados em laboratório utilizaram um hardware de alta performance em uma topologia de rede em que diversos clientes realizavam consultas sucessivas a um

servidor de nomes autoritativo, a fim de conhecer os limites deste servidor. O software DNS/ENUM escolhido para a avaliação foi software BIND, que é de fato um dos softwares mais conhecidos e utilizados hoje na Internet. Os resultados obtidos demonstraram que o software BIND teve um desempenho satisfatório e até mesmo surpreendente, principalmente quando comparado aos resultados apresentados em trabalhos similares, como os indicados em [19]. O software BIND alcançou uma vazão acima de 300 mil consultas por segundo para os dois grupos de registros avaliados (500k e 5M de registros), com um tempo de resposta abaixo de 1 ms. Estes resultados comprovam que o software BIND possui ótima escalabilidade com relação ao tamanho da base de dados de registros NAPTR, uma vez que a vazão de respostas às consultas não foi afetada negativamente, mesmo com um grupo de registros 10 vezes maior que o primeiro. Este ótimo desempenho se deve aos investimentos realizados em hardware e software que possibilitaram que a vazão máxima do servidor fosse aproximadamente 50 vezes maior que a vazão indicada em [19]. Esta constatação gera boas expectativas para o futuro do protocolo ENUM, uma vez que os desafios para a sua implantação vão sendo gradualmente superados com os avanços tecnológicos.

Após a realização dos testes laboratoriais, foram propostas duas estratégias de modelagem para o comportamento do servidor de nomes autoritativo. A primeira foi baseada na regressão de funções e a segunda abordagem utilizou as amostras obtidas nos testes em laboratório de forma direta. A modelagem baseada na utilização direta das amostras mostrou-se mais promissora, visto que minimiza o erro final do modelo ao utilizar os próprios valores reais dos parâmetros considerados, ao invés de um valor aproximado, oferecido pela abordagem baseada na regressão. Entretanto, o modelo por amostras possui limitações, uma vez que depende do acesso aos dados das medições realizadas em laboratório. Neste contexto, embora o modelo por regressão de funções seja um pouco menos exato, esta estratégia possibilita a predição de cenários que não foram contemplados nos testes em laboratório. Dessa forma, a plataforma de simulação utiliza as duas abordagens, dependendo do cenário em estudo.

O modelo de simulação foi implementado no ambiente de simulação NS-3, que por sua vez mostrou-se ser uma boa escolha, devido ao seu desempenho quanto à escalabilidade e aspectos relativos a manutenção, suporte, documentação e familiarização. Nesse sentido, ao considerar um sistema real simples, o modelo de simulação pôde ser bastante detalhista.

Dessa maneira, foram descritos três módulos principais: dois módulos de simulação e um módulo de configuração. O módulo de simulação *DnsQueryTool* é uma versão adaptada da ferramenta de testes *dnstperf*, que por sua vez é geralmente empregada para a emulação de clientes realizando consultas a um servidor de nomes autoritativo. Este módulo, apesar de mais simples que a ferramenta original, contempla todas as funcionalidades utilizadas nos testes em laboratório. Já o módulo de simulação *DnsAuthoritativeServer* é responsável por modelar o servidor de nomes autoritativo. Este módulo é baseado no software BIND e utiliza a mesma estrutura de banco de dados baseado em *zone files*. O desempenho deste servidor simulado pelo módulo *DnsAuthoritativeServer* tem como base os modelos matemáticos citados anteriormente. O módulo de configuração *DnsConfigTool* foi idealizado para facilitar o uso da plataforma de simulação por parte do usuário, ao criar os arquivos de configuração de forma transparente. Destaca-se ainda que, a implementação das mensagens de consulta e resposta DNS foi realizada de forma que estas mensagens tivessem o mesmo formato e tamanho das mensagens utilizadas no plano real. Esta abordagem realista, característica do NS-3, foi um aspecto importante para aumentar a confiabilidade do simulador.

Os resultados obtidos via simulação ao empregar a estratégia de modelagem baseada na utilização direta das amostras obtidas nos testes em laboratório foram superiores àqueles resultados obtidos ao utilizar-se a modelagem baseada na regressão de funções. Os resultados da simulação mostram-se também satisfatórios com relação ao erro cometido no processo de modelagem, validando assim a plataforma de simulação proposta.

Neste contexto, é possível destacar as principais contribuições do estudo apresentado nesta dissertação, relacionadas a seguir:

- Apresentação de resultados reais e atualizados do desempenho de um servidor DNS/ENUM baseado no software BIND. Os experimentos realizados em laboratório contribuem para que os servidores DNS/ENUM sejam corretamente implantados em um sistema real. Além disso, os resultados encontrados atualizam parcialmente trabalhos importantes, como o apresentado em [19].
- O estudo do modelo para o servidor de nomes autoritativo resultou em uma proposta de extensão da modelagem apresentada em [2] que é baseada na regressão de funções. Além disso, uma nova estratégia de modelagem baseada na utilização direta das amostras obtidas nos testes em laboratório foi proposta.

- A plataforma de simulação foi construída para ser integrada ao ambiente de simulação NS-3. Novos módulos e classes foram desenvolvidos para este ambiente, oferecendo novas possibilidades de trabalho para a comunidade acadêmica.

Além dessas contribuições diretas, existem ainda contribuições indiretas, como aquelas derivadas dos próprios resultados apresentados que poderão ser utilizados em outros trabalhos.

A plataforma de simulação proposta foi baseada em um sistema real simples. Contudo, o projeto da plataforma foi concebido de forma a possibilitar a futura inclusão de modelos baseados em sistemas mais complexos. Dentre as ações futuras mais imediatas para o trabalho apresentado nesta dissertação tem-se:

- Inclusão de modelos mais complexos que explorem diferentes arquiteturas e tipos do protocolo ENUM;
- Inclusão de modelos para outros softwares de servidores DNS/ENUM;
- Submissão dos módulos desenvolvidos para serem integrados ao código oficial do ambiente de simulação NS-3;
- Desenvolvimento de interface gráfica para facilitar a experiência de usabilidade do usuário;
- Inclusão de modelos para dispositivos de segurança.

Por fim, considera-se que a plataforma de simulação foi desenvolvida satisfatoriamente. Os resultados obtidos via simulação foram próximos daqueles encontrados em um sistema real. Espera-se que com o passar do tempo novas funcionalidades e modelos sejam disponibilizados, possibilitando o avanço dos estudos relacionados ao protocolo ENUM.

Referências Bibliográficas

- [1] N. Beijar, “Telephony Routing with Support for Number Portability in Interconnected Circuit and Packet Switched Networks,” Master’s thesis, Helsinki University of Technology, Espoo - Finland, 2004.
- [2] S. Balakrichenan, *An Autonomic Simulation Platform for Studying and Optimizing Addressing Issues in Next Generation Networks*. PhD thesis, Université Pierre et Marie Curie, 2008.
- [3] “Getting Underway with ENUM: Building a DNS Infrastructure for the Convergence of Telecommunications Networks.” White Paper, 2008.
- [4] R. A. C. Ribeiro, “Entraves à adesão do ENUM em Portugal,” Master’s thesis, Universidade Técnica de Lisboa, 2010.
- [5] S. Bradner, L. Conroy, and K. Fugiwara, “The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM).” RFC 6116, 2011.
- [6] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, “DNS Performance and the Effectiveness of Caching,” *Networking, IEEE/ACM Transactions on*, vol. 10, pp. 589–603, oct 2002.
- [7] J. Hwang, M. Mueller, G. Yoon, and J. Kim, “Analyzing ENUM Service and Administration from the Bottom Up: The Addressing System for IP Telephony and Beyond,” *CoRR*, vol. cs.CY/0109044, 2001.
- [8] B. Benyo, F. Hatwagner, T. Heckenast, K. Kovacs, A. Varga, and N. Varjasi, “Novel communication services based on ENUM technology,” in *Intelligent Engineering Sys-*

- tems, 2005. INES '05. Proceedings. 2005 IEEE International Conference on*, pp. 217–220, 2005.
- [9] “Evaluation of the Australian ENUM Trial,” tech. rep., Australian ENUM Discussion Group, 2007.
- [10] A. T. Jammulamadaka and M. A. Gregory, “Infrastructure ENUM Implementation in Australia,” in *TENCON 2010 - 2010 IEEE Region 10 Conference*, pp. 2304–2309, 2010.
- [11] A. T. Jammulamadaka, “Infrastructure Electronic Numbering Implementation in Australia,” Master’s thesis, RMIT University, 2010.
- [12] K. Reichinger, “ENUM in Austria: From the First Public Consultation to the National ENUM Field Trial.” Disponível em: <http://goo.gl/1dE5V>, 2002.
- [13] K. Reichinger, “ENUM in Austria: From High Expectations to Low Usage Figures. And Then?.” Disponível em: <http://goo.gl/BCqAx>, 2009.
- [14] “ENUM in the Netherlands,” tech. rep., Dutch ENUM group (NLEG), 2002.
- [15] L. A. Maris, “Infrastructure ENUM: Implementation options for the Netherlands,” Master’s thesis, Eindhoven University of Technology, 2006.
- [16] “The Irish ENUM Forum: Final Report,” tech. rep., PA Consulting Group, 2005.
- [17] “Japan ENUM Study Group Report,” tech. rep., Japan ENUM Study Group, 2003.
- [18] S. Yahya, S. E. A. Samah, and T. B. H. T. Besar, “Electronic Number Mapping in Malaysia – Design Framework,” in *Information Technology, 2008. ITSIM 2008. International Symposium on*, pp. 1–4, 2008.
- [19] C. Shen and H. Schulzrinne, “Measurement and Evaluation of ENUM Server Performance,” in *Communications, 2007. ICC '07. IEEE International Conference on*, pp. 1967–1972, 2007.
- [20] C. Shen and H. Schulzrinne, “Measurement and Evaluation of ENUM Server Performance,” Tech. Rep. cucs-029-06, Columbia University, 2006.

- [21] J. Rudinsky, “Private ENUM based Number Portability Administrative System Evaluation,” in *Ultra Modern Telecommunications Workshops, 2009. ICUMT '09. International Conference on*, pp. 1–7, 2009.
- [22] S. Balakrichenan, T. Bugnazet, and M. Becker, “Studying ENUM Performance with Modeling and Simulation,” in *Modelling Simulation, 2007. AMS '07. First Asia International Conference on*, pp. 82–87, 2007.
- [23] S. Balakrichenan, T. Bugnazet, and M. Becker, “Studying Global DNS Performance for ENUM Protocol,” in *Performance Modeling and Evaluation of Heterogenous Networks International Working Conference*, 2006.
- [24] S. Balakrichenan and M. Becker, “An Autonomic ENUM Implementation in Network Simulator-2,” in *Simutools '08 First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems and Workshops*, 2008.
- [25] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach*. Addison-Wesley, 5^a ed., 2010.
- [26] L. Dostálek and A. Kabelová, *Dns in Action: A Detailed And Practical Guide to Dns Implementation, Configuration, And Administration*. From technologies to solutions, Packt Publishing, 2006.
- [27] ITU-T, “The international public telecommunication numbering plan,” tech. rep., 2005.
- [28] T. Berners-Lee, R. Fielding, and L. Masinter, “Uniform Resource Identifiers (URI): Generic Syntax.” RFC 2396, 1998.
- [29] M. Mealling, “RFC 3403 Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database.” RFC 3403, 2002.
- [30] A. Johnston, *Sip: Understanding the Session Initiation Protocol*. Artech House telecommunications library, Artech House, 3^a ed., 2009.
- [31] M. Mealling, “Dynamic Delegation Discovery System (DDDS) Part Two: The Algorithm.” RFC 3402, 2002.

- [32] R. Stastny, “Introduction to ENUM Technical and Operational Aspects.” Disponível em: <http://goo.gl/k10Su>, 2001.
- [33] S. H. da Mata, P. R. Guardieiro, A. Cardoso, E. Lamounier, and L. C. Theodoro, “Performance Study of ENUM Service Based in Modeling and Simulation,” in *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*, pp. 1–5, 2011.
- [34] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*. Springer, 2011.
- [35] A. M. Law, “How to Build Valid and Credible Simulation Models,” in *Simulation Conference, 2008. WSC 2008. Winter*, pp. 39–47, 2005.
- [36] E. Weingartner, H. V. Lehn, and K. Wehrle, “A Performance Comparison of Recent Network Simulators,” in *Communications, 2009. ICC '09. IEEE International Conference on*, pp. 1–5, 2009.
- [37] K. Wehrle, M. Güneş, and J. Gross, *Modeling and Tools for Network Simulation*. Springer, 2010.
- [38] G. F. Riley, “Large-scale network simulations with GTNetS,” in *Simulation Conference, 2003. Proceedings of the 2003 Winter*, pp. 676–684 Vol.1, 2003.
- [39] P. E. Miyagi, *Introdução a Simulação Discreta*. Escola Politécnica da Universidade de São Paulo, 2006.
- [40] H. Perros, *Computer Simulation Techniques : The definitive introduction!* NC State University, 2007.
- [41] S. H. da Mata and P. R. Guardieiro, “Uma Introdução ao Protocolo ENUM,” in *IX Conferência de Estudos em Engenharia Elétrica*, 2011.
- [42] ISC, “BIND homepage.” Disponível em: <http://www.isc.org/software/bind>, 2010.
- [43] Nominum, “The dnsperf package.” Disponível em: <ftp://ftp.nominum.com/pub/nominum/dnsperf>, 2010.

- [44] Nominum, *dnsperf: DNS Performance Tool Manual*, 2007.
- [45] Volker Gropp, “bwm-ng homepage.” Disponível em: <http://www.gropp.org/?id=projects&sub=bwm-ng>, 2012.
- [46] K. Ramachandran and C. Tsokos, *Mathematical Statistics With Applications*. Academic Press, 2009.
- [47] T. Ryan, *Modern Regression Methods*. Wiley Series in Probability and Statistics, Wiley, 2008.
- [48] Furukawa, *Especificação Técnica PATCH CORD U/UTP GIGALAN CAT.6 ANATEL*, 2011.
- [49] NS-3 Project, *ns-3: NS-3 Manual*, 2011.
- [50] G. J. A. M. Carneiro, “NS-3: Network Simulator 3.” UTM Lab Meeting INESC Universidade do Porto, 2010.
- [51] R. Aitchison, *Pro DNS and Bind 10*. Pro to Expert Series, Apress, 2011.
- [52] The Wireshark Team, “Wireshark homepage.” Disponível em: <http://www.wireshark.org/>, 2011.

Apêndice A

Dados dos Testes em Laboratório

Este apêndice apresenta algumas tabelas com os resultados dos testes realizados em laboratório, de acordo com a descrição realizada no Capítulo 4.

Tabela A.1: Dados dos testes realizados em laboratório (500k registros).

Clientes	Consultas Perdidas (%)	Vazão (cps)	Tempo Médio de Resposta (s)	Desvio Padrão (s)	Utilização CPU (dnsperf) (%)	Utilização do Enlace de Comunicação (%)	Utilização da CPU (Bind)
2	0.0	6886	0.00029	4.2e-05	0.41	1.61	3.36
4	0.0	11356	0.000352	7.2e-05	0.75	2.65	5.8
6	0.0	19587	0.000304	0.000114	1.31	4.58	8.77
8	0.0	26601	0.000295	0.000131	1.67	6.21	11.31
10	0.0	32238	0.00031	0.000143	2.23	7.53	13.18
12	0.0	37736	0.000315	0.000155	2.55	8.82	14.68
14	0.0	43289	0.000317	0.000161	2.85	10.11	16.1
16	0.0	48634	0.000328	0.000165	3.33	11.36	17.46
18	0.0	54148	0.000329	0.000167	3.6	12.65	18.8
20	0.0	59242	0.000338	0.000171	4.01	13.84	20.17
50	0.0	129284	0.000387	0.000217	5.99	30.2	39.2
100	0.0	218279	0.000458	0.000435	8.28	50.99	58.92
200	0.0	304913	0.000653	0.001319	10.56	71.22	72.5
300	0.0	321633	0.000884	0.001913	11.14	77.63	75.46
Continua na próxima página							

Tabela A.1 – continuação da pagina anterior

Clientes	Consultas Perdidas (%)	Vazão (cps)	Tempo Médio de Resposta (s)	Desvio Padrão (s)	Utilização CPU (dnsperf) (%)	Utilização do Enlace de Comunicação (%)	Utilização da CPU (Bind)
400	0.006	309554	0.000923	0.002098	11.32	75.95	74.38
600	0.018	321408	0.000891	0.001557	11.25	78.44	75.05
800	0.031	302866	0.000928	0.001738	10.99	75.6	73.5
1000	0.044	307267	0.000933	0.001728	11.08	75.29	73.39

Tabela A.2: Dados dos testes realizados em laboratório (5M registros).

Clientes	Consultas Perdidas (%)	Vazão (cps)	Tempo Médio de Resposta (s)	Desvio Padrão (s)	Utilização CPU (dnsperf) (%)	Utilização do Enlace de Comunicação (%)	Utilização da CPU (Bind)
2	0.0	7847	0.000255	3.5e-05	0.46	1.83	3.54
4	0.0	11010	0.000363	7.3e-05	0.74	2.57	5.88
6	0.0	19997	0.000297	0.00011	1.36	4.67	9.01
8	0.0	26799	0.000293	0.000128	1.71	6.26	11.65
10	0.0	31943	0.000313	0.000145	2.25	7.46	13.51
12	0.0	37507	0.000317	0.000156	2.55	8.76	15.11
14	0.0	42766	0.00032	0.000162	2.86	9.99	16.53
16	0.0	48053	0.000332	0.000166	3.45	11.22	17.92
18	0.0	53258	0.000335	0.000171	3.66	12.44	19.34
20	0.0	59121	0.000338	0.000172	4.02	13.81	20.9
50	0.0	132683	0.000377	0.000208	6.14	30.99	42.62
100	0.0	220236	0.000454	0.000489	7.9	51.44	61.44
200	0.0	340760	0.000584	0.001441	11.84	79.6	80.23
300	0.0	361401	0.000805	0.00165	12.28	85.67	83.46
Continua na próxima página							

Tabela A.2 – continuação da pagina anterior

Clientes	Consultas Perdidas (%)	Vazão (cps)	Tempo Médio de Resposta (s)	Desvio Padrão (s)	Utilização CPU (dnsperf) (%)	Utilização do Enlace de Comunicação (%)	Utilização da CPU (Bind)
400	0.005	338625	0.00084	0.002149	12.33	84.51	82.51
600	0.016	338089	0.000848	0.001818	12.37	85.49	83.09
800	0.028	332701	0.000858	0.001929	12.13	84.26	82.17
1000	0.038	341941	0.000851	0.001739	12.31	85.12	82.22

Apêndice B

Conceitos Matemáticos

Este apêndice apresenta algumas definições que auxiliam o leitor na compreensão de conceitos matemáticos apresentados na dissertação.

B.1 O Método dos Mínimos Quadrados

Um modelo para regressão linear simples é dado pela equação,

$$y = \beta_0 + \beta_1 x + \varepsilon$$

onde β_0 é o interceptor do eixo y , β_1 é a inclinação da reta e ε é uma variável aleatória que representa o erro na predição da variável dependente. A equação de predição para este modelo é definida como,

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x.$$

onde $\hat{\beta}_0$ e $\hat{\beta}_1$ são conhecidos como estimadores da equação.

Considere, então, um valor observado (x_i, y_i) para o qual é possível obter um valor estimado \hat{y}_i dado por:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i.$$

A partir do valor estimado \hat{y}_i é possível encontrar o erro e_i definido por

$$e_i = (y_i - \hat{y}_i) = [y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)]$$

e, conseqüentemente, a soma dos quadrados dos erros (SQE) dada por

$$SQE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n [y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)]^2.$$

Logo, o Método dos Mínimos Quadrados consiste em encontrar os estimadores $\hat{\beta}_0$ e $\hat{\beta}_1$ que reduzem a SQE ao seu valor mínimo.

As expressões para $\hat{\beta}_0$ e $\hat{\beta}_1$ que reduzem a SQE ao seu valor mínimo são apresentadas a seguir:

$$\hat{\beta}_1 = \frac{S_{xx}}{S_{xy}} \quad (\text{B.1})$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad (\text{B.2})$$

onde,

$$S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2 \quad (\text{B.3})$$

$$S_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}). \quad (\text{B.4})$$

O leitor interessado pode encontrar a demonstração para as expressões acima em [46].

B.2 Teste *z-score* Modificado

O teste *z-score* modificado é utilizado para a identificação de *outliers* em um conjunto de dados para os quais será realizada uma regressão. Estes pontos com comportamento diferente do padrão apresentado pelos demais pontos podem afetar de forma drástica a equação estimadora resultante do processo de regressão.

O teste *z-score* modificado é baseado no teste *z-score*. O teste *z-score* utiliza a média e o desvio padrão como métricas para a identificação de *outliers*. Entretanto, uma vez que estas métricas são influenciadas pelos *outliers*, este método não é muito confiável. O teste *z-score* modificado, por sua vez, utiliza a mediana do desvio absoluto em relação a mediana dos dados observados, ou seja, utiliza a MAD (*Median Absolute Deviation*), dada por

$$MAD = \text{mediana}(|x_i - m|)$$

onde m é a mediana dos dados observados. Assim, o teste z -score modificado é definido como

$$z_i = \frac{(x_i - \bar{x})}{MAD}.$$

Um dado observado é considerado um *outlier* se o valor do teste z -score modificado for maior que 3,5.

B.3 Coeficiente de Determinação

O coeficiente de determinação (R^2) é uma medida de quão bem a equação de predição (equação B.1) estima o valor de y . O coeficiente de determinação é definido como:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

O valor de R^2 varia entre 0 e 1. Quanto mais próximo de 1 melhor é a qualidade da regressão.

B.4 Coeficiente de Correlação

O coeficiente de correlação é utilizado para mensurar o quanto duas variáveis aleatórias, por exemplo X e Y , estão linearmente relacionadas. O coeficiente de correlação entre X e Y é definido como,

$$r_{xy} = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}}$$

Os possíveis valores de r encontram-se entre -1 e 1 . Quanto mais próximo de -1 ou 1 o valor de r está, mais forte é a relação de linearidade das variáveis em questão.

B.5 Intervalo de Confiança para β_0 e β_1

Considere n pontos observados $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. A partir destes valores, segue o procedimento para obter o intervalo de confiança para o interceptor β_0 e a incli-

nação β_1 .

1. Calcule S_{xx} , S_{xy} , S_{yy} , \bar{x} e \bar{y} , conforme as equações B.3 e B.4.
2. Calcule $\hat{\beta}_1$ e $\hat{\beta}_0$ utilizando as equações

$$\hat{\beta}_1 = S_{xy}/S_{xx}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$$

3. Calcule a SQE pela equação

$$SQE = S_{yy} - \hat{\beta}_1 S_{xy}$$

4. Calcule a MQE (Média dos Quadrados do Erros) utilizando a equação

$$MQE = SQE/(n - 2)$$

5. O intervalo de confiança $(1 - \alpha)100\%$ para β_1 é dado por

$$\left(\hat{\beta}_1 - t_{\alpha/2, n-2} \sqrt{\frac{MQE}{S_{xx}}}, \hat{\beta}_1 + t_{\alpha/2, n-2} \sqrt{\frac{MQE}{S_{xx}}} \right)$$

onde $t_{\alpha/2}$ é encontrado pela tabela da distribuição t considerando-se $(n - 2)$ graus de liberdade.

6. O intervalo de confiança $(1 - \alpha)100\%$ para β_0 é dado por

$$\left(\hat{\beta}_0 - t_{\alpha/2, n-2} \left[MQE \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right) \right]^{1/2}, \hat{\beta}_0 + t_{\alpha/2, n-2} \left[MQE \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right) \right]^{1/2} \right)$$

B.6 Intervalo de Predição

O intervalo de predição $(1 - \alpha)100\%$ para y é dado por

$$\hat{y} \pm t_{\alpha/2} \sqrt{\frac{SQE}{n - 2} \sqrt{\left[1 + \frac{1}{n} + \frac{(x - \bar{x})^2}{S_{xx}} \right]}}$$

onde $t_{\alpha/2}$ é encontrado pela tabela da distribuição t considerando-se $(n - 2)$ graus de liberdade.

Apêndice C

Dados dos Testes de Simulação

Este apêndice apresenta algumas tabelas com os resultados dos testes de simulação, de acordo com a descrição realizada no Capítulo 6.

Tabela C.1: Dados dos testes de simulação, considerando-se o modelo baseado na regressão de funções (500k registros).

Clientes	Consultas Perdidas (%)	Vazão (cps)	Tempo Médio de Resposta (s)	Desvio Padrão (s)
2	0.02	6098	0.000327431	0.000150575
4	0.0	12117	0.000329612	0.000153483
6	0.0	18045	0.000332003	0.00015641
8	0.0	23901	0.000334212	0.000159362
10	0.0	29672	0.000336517	0.000162345
12	0.0	35355	0.000338914	0.000165385
14	0.0	40963	0.00034128	0.000168508
16	0.0	46483	0.000343719	0.000171661
18	0.0	51935	0.000346091	0.000174889
20	0.0	57305	0.000348513	0.000178187
50	0.0	129060	0.00038693	0.000233877
100	0.0	216805	0.000460765	0.00035969
Continua na próxima página				

Tabela C.1 – continuação da pagina anterior

Clientes	Consultas Perdidas (%)	Vazão (cps)	Tempo Médio de Resposta (s)	Desvio Padrão (s)
200	0.001	293372	0.000653139	0.000793082
300	0.002	296915	0.000926549	0.001650126
400	0.009	290176	0.000906785	0.001238148
600	0.026	294672	0.000918265	0.00123107
800	0.047	279089	0.000929214	0.001223486
1000	0.073	280707	0.000939648	0.001215278

Tabela C.2: Dados dos testes de simulação, considerando-se o modelo baseado na utilização direta de amostras (500k registros).

Clientes	Consultas Perdidas (%)	Vazão (cps)	Tempo Médio de Resposta (s)	Desvio Padrão (s)
2	0.0	6845	0.000291648	4.2007e-05
4	0.0	11296	0.000353604	7.2017e-05
6	0.0	19597	0.000305669	0.000113678
8	0.0	26925	0.00029662	0.000129732
10	0.0	32039	0.000311618	0.00014127
12	0.0	37839	0.000316636	0.00015235
14	0.0	43867	0.00031865	0.000157828
16	0.0	48456	0.000329703	0.000161875
18	0.0	54352	0.000330677	0.000163724
20	0.0	58794	0.000339675	0.000167706
50	0.0	128457	0.000388744	0.000210522
Continua na próxima página				

Tabela C.2 – continuação da pagina anterior

Clientes	Consultas Perdidas (%)	Vazão (cps)	Tempo Médio de Resposta (s)	Desvio Padrão (s)
100	0.0	217139	0.000460061	0.000381146
200	0.0	292238	0.000655679	0.000898356
300	0.002	310073	0.000887215	0.001268326
400	0.009	298301	0.000925681	0.001363796
600	0.025	309565	0.000894019	0.001121008
800	0.047	294715	0.000931096	0.001219178
1000	0.074	295719	0.000935621	0.001217182