

UNIVERSIDADE FEDERAL FLUMINENSE
CENTRO TECNOLÓGICO
MESTRADO EM ENGENHARIA DE TELECOMUNICAÇÕES

MICHELE PERPETUO CHEQUETTO HEMERLY BASTOS

SCGT – UM CONCEITO DE GERAÇÃO DE TRÁFEGO DISTRIBUÍDO

NITERÓI
2008

MICHELE PERPETUO CHEQUETTO HEMERLY BASTOS

SCGT – UM CONCEITO DE GERAÇÃO DE TRÁFEGO DISTRIBUÍDO

Dissertação apresentada ao Curso de Mestrado em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Comunicação de Dados Multimídia

Orientador: Prof^o Carlos Alberto Malcher Bastos, D.sc

Niterói

2008

**Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de
Computação da UFF**

B327 Bastos, Michele Perpetuo Chequetto Hemerly.
SCGT- um conceito de geração de tráfego distribuído/
Michele Perpetuo Chequetto Hemerly Bastos.- Niterói, RJ :
[s.n.], 2008.
161 f.

Orientador: Carlos Alberto Malcher Bastos
Dissertação (Mestrado em Telecomunicações) - Universidade
Federal Fluminense, 2008.

1. Redes de telecomunicação. 2 Geradores de tráfego. 3 Fluxo de
tráfego. 4.Desempenho de redes. I. Título.

CDD 621.382

MICHELE PERPETUO CHEQUETTO HEMERLY BASTOS

SCGT: UM CONCEITO DE GERAÇÃO DE TRÁFEGO DISTRIBUÍDO

Dissertação apresentada ao Programa de Mestrado em Engenharia de Telecomunicações como requisito parcial para obtenção do título de Mestre. Área de Concentração: Processamento de Sinais e Comunicação de Dados Multimídia.

Aprovada em 04 de Julho de 2008.

BANCA EXAMINADORA

Prof. Carlos Alberto Malcher Bastos - Orientador
UFF - Universidade Federal Fluminense

Prof.^a Maria Luiza D'Almeida Sanches
UFF - Universidade Federal Fluminense

Prof. Anilton Sales Garcia
UFES - Universidade Federal do Espírito Santo

A Deus, pelas maravilhas que tem feito em minha vida.

Aos meus pais, pelas orações, incentivos e apoio nos estudos desde minha infância.

Ao meu marido Paulo Sergio e meus pequeninos Gabriel e Tiago, pelo amor, alegria e paz no lar.

Ao meu orientador, Prof. Dr. Carlos Alberto Malcher Bastos, pela forma incentivadora e amiga, nos melhores e piores momentos, primando sempre pela qualidade e correção deste trabalho.

*“Graças se rendam a Deus, que nos dá a vitória por
nosso Senhor Jesus Cristo!”*

Paulo, discípulo de Jesus Cristo, in 1Cor 15, 57.

RESUMO

A crescente demanda por ambientes de medição ativa em redes tem motivado diversos grupos de pesquisa a envidar esforços na criação e adaptação de ferramentas, que utilizem um formato para a representação de informações através do uso de Serviços *Web*. Apesar do grande número de ferramentas existentes para a geração dos fluxos de tráfego, pouco trabalho foi realizado nesta direção. Este trabalho apresenta o sistema SCGT (Sistema de Coordenação para Geradores de Tráfego Sintético), que consiste em uma proposta de medição ativa, integrando geradores de tráfego, com as mais variadas características. Este sistema disponibiliza, através do Serviço *Web*, medidas relacionadas ao desempenho de redes em alta velocidade, permitindo gerar tráfegos simultâneos e agendados, através de uma ou mais estações de trabalho. Além disso, fornece uma interface bastante amigável para os usuários finais.

Palavras-chave: Fluxos de Tráfego. Geradores de Tráfego. Desempenho de Redes.

ABSTRACT

The growing demand for environment of active measurement for computer networks, motivated several research groups to endeavor in the development and adaptation of tools, which format the representation of information through the use of Web Services. In spite of the existent of a great number of traffic flow generation tools, little work was carried out in this direction. This work presents the SCGT (System of Coordination Generation for Synthetic Traffic) system, which consists of a solution for active measurement, integrating traffic generators, with the most varied characteristics. This system is available, through the Service Web, with measures related to performance of high speed networks, allowing the generation of simultaneous and scheduled traffics, through one or more stations of work. Beside that, it supplies a quite friendly interface for the final users.

Keywords: Traffic Flow. Traffic Generation. Networks Performance.

LISTA DE QUADROS, GRÁFICOS E TABELAS

Quadro 1: Atraso Unidirecional	36
Quadro 2: Características de Funcionamento	55
Quadro 3: Tipos de tráfego	56
Quadro 4: Parâmetros de entrada (UDP)	56
Quadro 5: Parâmetros de entrada (TCP)	57
Quadro 6: Informações de saída	57
Quadro 7: Descrição do teste de Transferência máxima	92
Quadro 8: Descrição do teste de DNS	94
Quadro 9: Descrição do teste de conectividade e simultaneidade	98
Quadro 10: Descrição do teste Unidirecional	103
Tabela1: Comparação dos Projetos de Medição Ativa	61
Gráfico 1: Taxa de Recebimento em função do tamanho do pacote	93
Gráfico 2: DNS – <i>Bitrate x Time</i>	95
Gráfico 3: DNS – <i>Delay x Time</i>	96
Gráfico 4: DNS – <i>Jitter x Time</i>	96
Gráfico 5: DNS – <i>Packet x Time</i>	97
Gráfico 6: Diferença dos Valores de <i>Offset</i>	98
Gráfico 7: Simultâneo - <i>packet loss x tempo</i> – primeiro segundo	99

Gráfico 8: Simultâneo - <i>packet loss</i> x tempo – segundo sentido	100
Gráfico 9: Simultâneo - <i>delay</i> x tempo – primeiro sentido	100
Gráfico 10: Simultâneo - <i>delay</i> x tempo – segundo sentido	101
Gráfico 11: Simultâneo - <i>Bitrate</i> x tempo – primeiro sentido	101
Gráfico 12: Simultâneo - <i>Bitrate</i> x tempo – segundo sentido	102
Gráfico 13: Retardo Unidirecional Máquina 1x Servidor de NTP	104
Gráfico 14: Retardo Unidirecional Máquina 2x Servidor de NTP	104

LISTA DE FIGURAS

Figura 1: Sentido do fluxo	34
Figura 2: Hora fornecida por uma autoridade externa	40
Figura 3: Modo de Funcionamento do algoritmo de Cristian	41
Figura 4: Troca de mensagens do algoritmo de Cristian	41
Figura 5: Estrutura Hierárquica do NTP	43
Figura 6: Mapa dos Servidores <i>Skitter</i>	60
Figura 7: Integração dos Processos	66
Figura 8: Modo de Comunicação	67
Figura 9: Geração do Tráfego Programada	69
Figura 10: Processos Cliente e Servidor – Instantâneo	71
Figura 11: Processos Cliente e Servidor – Agendado	71
Figura 12: Configuração da Rotina de Cron	71
Figura 13: Sincronismo NTP na rede GIGA	72
Figura 14: Arquitetura de Sincronismo Proposta para a rede GIGA	73
Figura 15: Diagramas de Casos de Uso	77
Figura 16: Diagramas de Classe	84
Figura 17: Diagrama de Seqüência – Ferramenta	85
Figura 18: Diagrama de Seqüência – Modo	85

Figura 19: Diagrama de Seqüência – Perfil IPERF	86
Figura 20: Diagrama de Seqüência – Perfil DITG	87
Figura 21: Diagrama de Sequencia – Finalizar	88
Figura 22: Diagrama de Seqüência – Relatório	88
Figura 23: Topologia do SCGT na rede GIGA	91
Figura 24: Taxa Gerada	93
Figura 25: Taxa de Transferência	94
Figura 26: Perfil DNS	95
Figura 27: IPERF x DITG – Medição Simultânea para mesmo sentido	99
Figura 28: Configuração Unidirecional	103
Figura 29: Resultado do primeiro fluxo	106
Figura 30: Resultado do Segundo Fluxo	106
Figura 31: Apresentação do SCGT	160
Figura 32: Perfil de Tráfego – TCP – Modo Agendado	161
Figura 33: Lista de Relatórios	161

LISTA DE ABREVIATURAS OU SIGLAS

AIEPM	<i>Active Internet End-to-End Performance Measurements</i>
AMP	<i>Active Measurement Program</i>
ANS	<i>Advanced Network Services</i>
ATM	<i>Asynchronous Transfer Mode</i>
BSD	<i>Berkeley System Distributed</i>
CAIDA	<i>Cooperative Association for Internet</i>
CWDM	<i>Coarse Wavelength Division Multiplexing</i>
DHTML	<i>Dynamic HTML</i>
DITG	<i>Distributed Internet Traffic Generator</i>
DNS	<i>Domain Name Server</i>
DWDM	<i>Dense Wavelength Division Multiplexing</i>
FLL	<i>Frequency Locked Loop</i>
GPS	<i>Global Position System</i>
GSG	<i>Common Solutions Group</i>
HTML	<i>Hyper Text Markup Language</i>
http	<i>Hyper Text Transfer Protocol</i>
ICMP	<i>Internet Connection Multiplexing Protocol</i>
IEEE	<i>Institute for Electrical and Electronics Engineers</i>

IETF	<i>Internet Engineering Task Force</i>
IPDV	<i>IP Packet Delay Variation</i>
IPPM	<i>IP Performance Metrics</i>
ITU-T	<i>International Telecommunications Union- Telecommunication Standardization Sector</i>
MGEN	<i>Multi Generator</i>
NLANR	<i>National Laboratory for Applied Network Research</i>
NRL	<i>National Research Laboratory</i>
NTP	<i>Network Time Protocol</i>
OWDM	<i>One Way Delay Measurement</i>
PDU	<i>Protocol Data Unit</i>
PHP	<i>Hypertext Preprocessor</i>
PLL	<i>Phase Locked Loop</i>
QoS	<i>Quality of Service</i>
RFC	<i>Request for Comments</i>
RTT	<i>Round Trip Time</i>
SSH	<i>Secure Shell</i>
TCP	<i>Transmission Control Protocol</i>
TMG	<i>Traffic Monitor Generator</i>
TTL	<i>Time-to-Live</i>
UDP	<i>User Datagram Protocol</i>
UML	<i>Unified Modeling Language</i>
UTC	<i>Coordinated Universal Time</i>
VAD	<i>Voice Active Detect</i>
VoIP	<i>Voice over Internet Protocol</i>

SUMÁRIO

Capítulo 1

Introdução	18
1.1. Objetivos	19
1.2. Contribuições	20
1.3. Estrutura do Trabalho	20

Capítulo 2

Revisão Bibliográfica	22
2.1. Projeto GIGA	22
2.2. Métricas segundo Organismos de Padronização	23
2.3. Sistemas Distribuídos	24
2.4. Ferramentas de Medição Ativa	24

Capítulo 3

Estudos de Métricas e Medidas	26
3.1. Introdução	26
3.2. Utilização da Medição e Geração de Tráfego	26
3.2.1. Medição Passiva	27
3.2.2. Medição Ativa	27

3.3. Medidas Relativas	28
3.3.1. Medições relativas à utilização de recursos	28
3.3.2. Medições de tráfego relativas ao desempenho	28
3.3.2.1. Medições Unidirecionais	28
3.3.2.2. Medições de Ida e Volta (<i>Round Trip Time</i>)	29
3.4. Métricas mais comuns segundo IEEE e ITU-T	29
3.4.1. Latência	30
3.4.2. Perda de Pacotes	31
3.4.3. Vazão	31
3.4.4. Capacidade	32
3.4.5. Utilização do Enlace	32
3.4.6. Banda Disponível	32
3.5. Métricas mais comuns segundo IPPM – IETF	32
3.5.1. Conectividade	33
3.5.2. Retardo Unidirecional	34
3.5.3. Retardo de Ida e Volta (<i>Round Trip Time</i>)	34
3.5.4. Variação do retardo	35
3.6. Comparativo entre Organismos de Padronização	35
3.7. Conclusão	36
Capítulo 4	
Sincronismo dos Relógios em Rede	37
4.1. Introdução	37
4.2. Alguns Conceitos em Medição na Sincronização de relógios em rede	38
4.3. Relógios de Alta Precisão	39
4.4. Sincronização de Relógios em Rede	40
4.5. Algoritmos de Sincronização Interna e Externa	41

4.5.1. Algoritmo de Cristian	41
4.5.2. Algoritmo de Berkley	42
4.5.3. NTP – Network Time Protocol	42
4.5.3.1. Arquitetura do Protocolo	42
4.5.3.2. Tipos de Associações	43
4.5.3.3. Métricas Relativas a qualidade de sincronização	44
4.5.3.4. Algoritmos de Sincronização	44
4.6. Conclusão	47
Capítulo 5	
Ferramentas para Geração e Medição de Tráfego	48
5.1. Introdução	48
5.2. Ferramentas de Geração e Medição	49
5.3. Avaliação do Software	49
5.4. Critérios de Comparação	50
5.5. Ferramentas Analisadas	50
5.5.1. TMG	50
5.5.2. MGEN	51
5.5.3. Rude/Crude	52
5.5.4. IPERF	52
5.5.5. DITG	53
5.6. Comparação das Ferramentas	54
5.6.1 Características Gerais das Ferramentas	55
5.7. Justificativa de Escolha das ferramentas	58
5.8. Projetos de Medição Ativa	59
5.8.1. CAIDA	59
5.8.1.1. Skitter	59

5.8.2. AMP – <i>Active Measurement Program</i>	60
5.8.3. <i>Surveyor</i>	61
5.9. Comparação dos Projetos Avaliados	61
5.10. Conclusão	62
Capítulo 6	
SCGT – Sistema de Coordenação para Geradores de Tráfego Distribuídos	63
6.1. Introdução	63
6.2. Descrição Geral	64
6.3. Serviços	64
6.4. Produtos	65
6.5. Composição do Sistema	65
6.6. Aspectos Relacionados ao Funcionamento	66
6.6.1. Modelo de Arquitetura	66
6.6.2. Divisão Estrutural	66
6.6.3. Perfil de Tráfego	67
6.6.4. Geração do Tráfego Programada	68
6.6.5. Rotina de Cron	69
6.6.6. Autenticação com OpenSSH	69
6.6.7. Ordenação dos Processos	70
6.6.7.1. Modo Instantâneo	70
6.6.7.2. Modo Agendado	71
6.7. Sincronismo nos Experimentos	71
6.7.1. Arquitetura Proposta	72
6.7.2. Problemas de Sincronismo para Tráfegos Simultâneos	73
6.8. Conclusão	73
Capítulo 7	

Análise e Projeto do SCGT	75
7.1. Introdução	75
7.2. Visão Geral	76
7.3. Requisitos do Sistema	76
7.3.1. Diagrama de Caso de Uso	77
7.4. Diagrama de Classes	83
7.4.1. Diagramas de Seqüência	84
7.5. Conclusão	88
Capítulo 8	
Testes	90
8.1. Introdução	90
8.2. Toplogia do SCGT	90
8.3. Descrição dos Testes	92
8.3.1. Primeira Etapa – Teste Individual	92
8.3.1.1. Taxa de Transferência Máxima	92
8.3.1.2. Aplicação DNS	94
8.3.2. Segunda Etapa – Testes Simultâneos	97
8.3.2.1. Sincronismo entre as Máquinas	97
8.3.2.2. Conectividade	98
8.3.2.2. Medidas Unidirecionais	102
8.4. Conclusão	107
Capítulo 9	
Conclusão e Trabalhos Futuros	108
9.1. Conclusão e Considerações Finais	108
9.2. Trabalhos Futuros	109

Capítulo 10

Referências Bibliográficas	110
10.1. Referências Bibliográficas	110
Apêndice A – Visão Tecnológica da Rede GIGA	114
Apêndice B – Código Fonte do Módulo CGT	118
Apêndice C – Relatórios e Processos	149
Apêndice D – Pré-Requisitos	156
Apêndice E – Operacionalidade	160

Capítulo 1

Introdução

Com o crescimento das redes IP e do uso destas para propósitos mais diversos, torna-se cada vez mais importante conhecer as suas características de desempenho. Para tal se faz necessário coletar dados sobre o comportamento da rede dos mais variados modos, onde estes são correlacionados e analisados para que seja possível determinar o desempenho e a qualidade dos seus serviços, bem como possíveis situações de anormalidades.

As atividades realizadas para a observação e coleta de dados na rede podem ser agrupadas em três grandes funções: caracterização do tráfego, monitoração da rede e controle do tráfego.

- Caracterização do tráfego e planejamento da capacidade: É feita através de algumas observações, como padrões de pico e suas variações estatísticas, incluindo o desenvolvimento de perfis de tráfego para capturar variações. Também é feita a determinação da distribuição do tráfego na rede, buscando auxiliar o planejamento da capacidade, juntamente com a estimativa da carga de tráfego nos diferentes roteadores e na rede. Adicionalmente, é possível observar as tendências de crescimento do tráfego de forma a prever uma futura demanda.
- Monitoração da rede: A determinação do estado operacional da rede, incluindo detecção de falhas, é uma necessidade básica para qualquer rede. Esta pode ser utilizada como um meio de levantar questões de desempenho apresentadas pelos clientes e avaliar a efetividade das políticas de engenharia de tráfego. Estas ações podem ser baseadas no uso dos dados históricos de desempenho.
- Controle do tráfego: A otimização adaptativa do desempenho da rede em resposta a eventos como, por exemplo, o re-roteamento para contornar congestionamentos ou falhas, é a principal utilização do controle de tráfego, sendo bastante importante para estudos de engenharia de tráfego.

Existem várias abordagens para observar e medir as características de uma rede. As duas abordagens mais comuns são a medição passiva e a medição ativa. Na medição passiva é observado o tráfego da rede sem interferências, enquanto na medição ativa é injetado um tráfego sintético na rede entre os diferentes pontos de medição.

A medição ativa é essencial para o planejamento e o gerenciamento de uma rede, onde a geração do tráfego sintético deve abranger uma ampla variedade de grandezas e de funções, cobrindo desde serviços básicos, até requisitos de desempenho da rede para aplicações específicas, tais como retardo unidirecional e latência. Entretanto, para esta medição poder ser utilizada com confiabilidade em uma ampla rede, o pré-requisito é a implantação de uma infraestrutura com alguns aspectos de bastante relevância quais sejam: os locais e as formas como estas medições são feitas, bem como as características do tráfego a ser analisado. Algumas medições ativas tomam como base observações feitas em um único ponto da rede. Outras tomam como base diferentes características de tráfego, observadas em mais de um ponto, correlacionando coletas feitas no mesmo horário, mesmo que sejam tomadas em poucos lugares distintos e para tal, a diferença entre os relógios pode ser de bastante importância para a obtenção de tais medidas.

Com o objetivo de unificar os experimentos de medição ativa em uma única base de dados padronizada, capaz de ser entendida e acessível através de uma interface *Web* [WIKI08], é proposto nesta dissertação um Sistema de Coordenação para Geradores de Tráfego Sintético (SCGT). Este sistema visa oferecer diferentes serviços de geração e coleta de tráfego para uma rede amplamente distribuída e que opera em alta velocidade, integrando algumas ferramentas de código aberto destinadas ao uso de medição ativa. Além disto, o SCGT visa agilizar e facilitar a interação do usuário com as ferramentas inicialmente propostas para esta primeira versão do sistema.

Ao fim da dissertação, o principal resultado é a construção de um sistema que possa atender de forma satisfatória as características exigidas para a rede GIGA [MART05], permitindo uma infra-estrutura de medição ativa e eficiente.

1.1. Objetivo

Mais especificamente esta dissertação aborda questões correlacionadas com a geração de tráfego para medições de desempenho para a rede GIGA, que opera com velocidades em Gbps. O principal objetivo deste trabalho é propor um sistema que possibilite a integração de diferentes perfis de ferramentas de código aberto para geração de tráfego sintético, e que estas sejam

capazes de operar simultaneamente, e de forma confiável, em um ambiente com características da rede GIGA.

1.2. Contribuições

Este trabalho contribui para a medição de desempenho da rede GIGA, proporcionando credibilidade e flexibilidade para a realização de coletas obtidas através desta, criando um sistema capaz de integrar e operar simultaneamente a geração de diferentes características de tráfego, utilizando a arquitetura *Web* [WIKI08] como referência. Tal fato se justifica por não ter sido encontrado em nenhuma das ferramentas destinadas ao uso de medições ativas, tais características, entendidas como essenciais para a implementação em um ambiente amplamente distribuído.

1.3. Estrutura do Trabalho

A estrutura desta dissertação se caracteriza por apresentar primeiramente, os conceitos e teorias envolvidas sobre medições ativas e aspectos relacionados à sincronização destas em um ambiente distribuído. Posteriormente é feita a apresentação e análise do sistema, e são descritos os resultados obtidos por meio deste.

O capítulo dois apresenta uma síntese dos principais trabalhos relacionados com o tema central da dissertação e que serviram de referência teórica para este trabalho.

No capítulo três, são apresentados estudos relacionados ao uso de métricas, para a realização de técnicas de medição em redes. Estas foram divididas em métricas tradicionalmente utilizadas, baseadas em padrões do *Institute for Electrical and Electronics Engineers - IEEE* [IEEE08] e do *International Telecommunications Union – ITU-T* [ITUT08], e métricas desenvolvidas pelo *Internet Engineering Task Force - IETF* [IETF08] para redes IP. As métricas e metodologias de medição descritas neste capítulo demonstram a importância da existência de métricas bem definidas, identificando os parâmetros de qualidade e desempenho de redes, bem como o esforço feito pelos organismos de padronização e pela comunidade internacional de pesquisa para o seu desenvolvimento.

No capítulo quatro, são retratados aspectos relacionados ao estudo de sincronismo de máquinas em rede, apresentando os principais conceitos em medição e sincronização de relógios, e os tipos de algoritmos existentes, dando ênfase ao *Network Time Protocol - NTP* [NTP08], por este ser um protocolo destinado a sincronização hierárquica em redes amplamente distribuídas.

No capítulo cinco são analisadas algumas ferramentas de código aberto, desenvolvidas pela comunidade internacional de pesquisa para a geração de tráfego sintético. Foram detalhadas suas várias características de acordo com a norma NBR 13596 [NBR96], fazendo um comparativo segundo suas funcionalidades, usabilidade, manutenibilidade e portabilidade, e selecionando as ferramentas mais adequadas para compor o sistema. Ao final do capítulo é apresentado um levantamento dos principais projetos internacionais que utilizam testes de medição ativa, com ênfase nos tipos de medições realizadas, e nas ferramentas atribuídas para seus funcionamentos. A seguir é feito um comparativo desses projetos entre si e provando o impacto destes, em aspectos relacionados às medições ativas através do uso *Web*.

No capítulo seis é descrito o Sistema de Coordenação para Geradores de Tráfego (SCGT), explicando os objetivos envolvidos nesta arquitetura, detalhando o modo de funcionamento e a interação deste em um ambiente destinado à medição ativa.

No capítulo sete é explorado o processo de Engenharia de Software, onde é feita uma análise mais específica de interação do sistema SCGT, explorando requisitos funcionais e não funcionais, além do detalhamento de diagramas.

O capítulo oito é o responsável por dar uma visão mais específica da implementação do SCGT, onde são descritos os testes relativos à operacionalidade do sistema, bem como os resultados obtidos através destes.

No capítulo nove é apresentada uma conclusão desta dissertação, discutindo sobre os resultados obtidos através do uso do Sistema SCGT, bem como das facilidades e contratempos encontrados durante o desenvolvimento do mesmo. Ao final, são apresentados trabalhos que poderão ser utilizados no futuro, ampliando e dando continuidade ao já realizado.

O apêndice A retrata a rede GIGA em vários aspectos, demonstrando sua infra-estrutura e os equipamentos que envolvem a rede.

O apêndice B detalha passo-a-passo o código fonte usado no SCGT.

O apêndice C relata como são feitos os processos para obtenção dos relatórios finais do SCGT.

O apêndice D detalha como é feita a instalação do SCGT nas máquinas.

Por fim, no apêndice E são demonstrados aspectos relacionados com a operacionalidade e manipulação do SCGT.

Capítulo 2

Revisão Bibliográfica

Este capítulo tem por objetivo apresentar uma síntese das principais referências bibliográficas estudadas durante o desenvolvimento da dissertação e que estão diretamente relacionadas com os objetivos centrais deste trabalho. Tais referências abordam aspectos como o modo de funcionamento e os componentes que envolvem a rede GIGA, as métricas mais utilizadas em redes de telecomunicações segundo os organismos de padronização internacional e os fundamentos principais que envolvem sistemas atuando de forma distribuída.

2.1. Projeto GIGA

O Projeto Giga [GIGA08] consiste no uso e implementação de uma rede óptica experimental voltada ao desenvolvimento de tecnologias, aplicações e serviços de telecomunicações associados à tecnologia IP e banda larga. Este projeto também prevê a transferência de tecnologia para empresas brasileiras.

A rede do Projeto Giga tem velocidade de transmissão de 2,5 Gbps podendo chegar até 10 Gbps. Abrangendo os municípios de Campinas, São Paulo, São José dos Campos, Cachoeira Paulista, Rio de Janeiro, Niterói e Petrópolis, a rede interconecta 17 universidades e centros de pesquisa do eixo Rio-São Paulo. O uso desta rede se dá através de subprojetos de pesquisa e desenvolvimento em quatro áreas temáticas: redes ópticas; serviços experimentais de telecomunicações; protocolos e serviços de rede; e serviços e aplicações científicas. Estes subprojetos são apresentados através do relatório “GigaIQoM – Infra-estrutura de Medições para a Rede Giga (Subprojeto 2465)” [IQOM08] que descreve o resumo das atividades realizadas dentro do escopo do subprojeto 2465 do Projeto Giga.

Atualmente a Universidade Federal fluminense – UFF, conta com um subprojeto de infra-estrutura para medições ativas. Este subprojeto consiste na análise e proposição de um sistema de geração de tráfego sintético para validar serviços em redes de alta velocidade. Para tal propósito, o artigo “Visão Tecnológica da Rede Experimental de Alta Velocidade” [MART05] foi de grande valia, esclarecendo detalhes da tecnologia usada no Projeto Giga, bem como são feitas as configurações dos equipamentos e os tipos de servidores utilizados na rede. Dentre os servidores, pode-se observar que a rede GIGA utiliza servidores NTP para o sincronismo entre as máquinas.

2.2.Métricas segundo Organismos de Padronização

Organismos de padronização são organizações nacionais ou internacionais que são reconhecidas como autorizadas a escrever padrões para serem utilizados na área de telecomunicações. Como existem diversos organismos de padronização que divulgam documentos semelhantes, nessa dissertação, são detalhadas métricas tradicionais utilizadas em redes de telecomunicações com diferentes enfoques, e que são apresentadas por organismos de padronização internacionais mais reconhecidos na área de telecomunicações.

O ITU *Telecommunications Standardization Sector* – ITU-T é um órgão permanente do *International Telecommunications Union* - ITU, responsável por estudar questões técnicas e operacionais de rede de telecomunicações e elaborar recomendações. Os trabalhos mais significativos desenvolvidos pelo ITU-T e estudados nesta dissertação, são as recomendações das séries E.400 a E.899, I.200 a I.250 e G.1000, que padronizam um conjunto de métricas úteis e claras para análise de desempenho de redes, de forma que estas possam ser de fácil entendimento e utilização.

O *Internet Engineering Task Force* – IETF [IETF08] é uma comunidade internacional ampla e aberta que tem como missão identificar e propor soluções a questões/problemas relacionados à utilização da *Internet*, além de propor padronização das tecnologias e protocolos envolvidos. O grupo de trabalho *IP Performance Metrics* [IPPM08] do IETF desenvolveu o “*Framework for IP Performance Metrics*” [RFC2330], como sendo um conjunto de definições, métricas e recomendações para medição de desempenho de redes IP. Este *Framework* apresenta termos para descrever a rede, explica a necessidade de métricas úteis, concretas e bem definidas, capazes de serem medidas de forma confiável. As recomendações mais utilizadas para as métricas do IETF foram baseadas segundo as RFC’s: 2678 [RFC2678] que relata aspectos relacionados aos tipos de conectividade existentes, 2679 [RFC2679] que apresenta aspectos relacionados ao uso de medições *One Way Delay Metric* – OWDM, 2680 [RFC2680] que

descreve de forma clara aspectos relacionados a perda de pacotes em medições OWDM e 2681 [RFC2681] que retrata medições relacionadas a atrasos utilizando *Round Trip Time* - RTT.

O estudo de métricas segundo organismos de padronização colabora para um maior entendimento sobre métricas que possam ser utilizadas para ambientes de testes de desempenho, sobretudo em uma rede operando em velocidade de gigabit por segundo.

2.3. Sistemas Distribuídos

Um sistema distribuído pode ser visto como uma coleção de computadores autônomos conectados por rede e equipados com *softwares* distribuídos.

A proposta desta dissertação é a criação de um sistema para atuar de forma distribuída na rede GIGA, permitindo experimentos simultâneos de medição ativa através do uso de vários computadores. Neste sentido, a literatura “Sistemas Distribuídos Conceitos e Projetos” [COUL08] foi utilizada para obter detalhes das diferentes formas de sincronização e concorrência dos processos operando de forma distribuída. Esta literatura abrange de forma resumida os serviços de sincronização existentes. Como manter a sincronização de relógios é uma tarefa importante e difícil em sistemas distribuídos, o artigo “Caracterização da Rede de Sincronização na Internet” [WPER08] descreve o *Network Time Protocol* – NTP que discorre sobre diversos aspectos que definem a qualidade do tempo, bem como as características topológicas da rede. Os resultados mostrados neste artigo, demonstram a evolução da sincronização via NTP nos últimos anos, e porque este é o serviço de sincronização mais confiável e utilizável até o momento.

Detalhes específicos sobre o protocolo NTP foram apresentados através das RFC’s 1305 [RFC1305] e 2030 [RFC2030] e serviram para validar o funcionamento do protocolo, além de obter informações dos algoritmos que o compõem.

2.4. Ferramentas de Medição Ativa

Medições ativas são importantes para uma otimização adaptativa e para determinar se está sendo oferecido certo nível de qualidade a determinadas aplicações. No entanto, para que estas sejam efetivas, devem ser obtidas e aplicadas de modo sistemático fazendo-se, necessário, portanto, a existência de uma infra-estrutura de medições.

Para a criação de um sistema de medição ativa para a rede GIGA, que pudesse funcionar de forma genérica e que tivesse certo nível de detalhamento nas medições (granularidade específica), foram investigados através da *Web* diversos esforços voltados para a implantação de

infra-estruturas de medições de grande porte. Na Internet2 [INT208], dentro da iniciativa de desempenho fim a fim (*End-to-End Performance Initiative*) é apresentado o desenvolvimento de um ambiente de melhoria do desempenho fim-a-fim para redes de grande porte, denominado de *End-to-End Performance Improvement Performance Environment System - E2E piPEs* [PIPE08], que se baseia no uso de medições ativas realizadas através de diversas ferramentas, sendo como uma das principais a IPERF [IPERF08] usada para medir a largura de banda disponível. A idéia desde ambiente é a de aproveitar medidas coletadas através de diversas ferramentas.

Na Europa existem também alguns projetos que serviram como exemplo para o desenvolvimento do SCGT, dentre estes a *Cooperative Association for Internet Datg e a Analysis – CAIDA* [CAID08] e o projeto *Surveyor* [SUR08]. No entanto, estes projetos se limitam a determinados tipos de medição e utilizam ferramentas bem conhecidas de medição ativa, como por exemplo, ping e traceroute.

A norma NBR13596 [NBR96] que trata de aspectos relacionados à qualidade e diretrizes para uso de um *software* foi utilizada neste trabalho como principal referência, para obter as características necessárias nas ferramentas de código aberto destinadas ao uso de medição ativa, de forma que estas pudessem compor o sistema SCGT.

Capítulo 3

Estudo de Medidas e Métricas relativas

3.1. Introdução

As redes IP possuem diversos problemas de infra-estrutura, tais como a falta de controle de erros e atrasos, congestionamento das filas dos equipamentos, falta de priorização de serviços, dentre outros. Em função disso é necessário se ter informações relevantes sobre a análise do tráfego, possibilitando a elaboração de diagnósticos mais ricos e precisos, facilitando assim o processo de tomada de decisão.

Analisar um tráfego permite, dentre outras coisas, a identificação das anomalias da rede em termos de segurança, ilustrando, por exemplo, as tentativas de ataques a uma determinada rede ou serviço.

Um dos recursos utilizados para a análise é a matriz de tráfego, que consiste em um resumo das demandas de tráfego entre diversos pontos da rede sob aspectos diversos. Uma matriz de tráfego pode estar no nível de rede, enlace, sub-rede, etc. Um outro benefício corresponde à obtenção de informações quanto aos pontos de gargalo na rede, e à tendência de congestionamento nos enlaces.

3.2. Utilização da Medição e Geração de Tráfego

No caso de redes com múltiplas administrações, como a Internet, as características de um caminho tais como a capacidade e a latência, podem não ser previamente conhecidas. Neste contexto, a única forma de monitorar um caminho que inclui várias redes, pode ser a utilização

de medições fim-a-fim, que requerem a cooperação apenas dos pontos terminais, ou mesmo de um único ponto do caminho.

3.2.1 Medição Passiva

Medições passivas são obtidas observando o tráfego normal da rede, ou seja, não afetam a rede. Essas medidas podem ser feitas utilizando-se equipamentos especialmente desenvolvidos para este propósito, como um *sniffer*, ou funções embutidas em outros equipamentos, como roteadores, *switches* ou mesmos equipamentos de acesso à rede e interface com o usuário. Os equipamentos são periodicamente acessados através de um *polling* realizado pela estação de gerenciamento e os dados coletados são transferidos para uma base de dados centralizada. Essas informações permitem então avaliar o desempenho e o estado da rede.

Na abordagem passiva é medido o tráfego real. Entretanto, o mecanismo de consulta necessário para coleta dos dados, e os alarmes e avisos enviados pelos equipamentos, geram tráfego na rede, que pode ser considerável. Além disso, a quantidade de dados obtidos pode ser substancial, especialmente quando é feita análise de fluxos e captura de informação de todos os pacotes.

Uma vez que, com esta abordagem, é possível a visualização de todos os pacotes na rede, podem surgir questões sobre a privacidade ou a segurança, da forma como são feitos o acesso e a proteção aos dados obtidos.

Uma desvantagem da medição passiva é o fato de que as medidas se baseiam no tráfego que flui através do enlace que está sendo medido. Caso este tráfego não possua o volume e/ou outras características necessárias para a medição passiva, existe a possibilidade de que as medidas obtidas não sejam confiáveis.

3.2.2. Medição Ativa

A medição ativa [RFC1262] se baseia na capacidade de injetar pacotes de teste na rede, ou enviar pacotes para servidores e aplicações observando seu progresso na rede. Uma vez que é criado tráfego extra, este tráfego e seus parâmetros são artificiais. O volume e outros parâmetros do tráfego introduzidos devem ser totalmente ajustáveis de forma que pequenos volumes de tráfego sejam suficientes para obter medidas significativas.

Por outro lado, a medição ativa permite o controle explícito da geração de pacotes para criar os cenários de medição. Isso inclui o controle da natureza do tráfego gerado, as técnicas de amostragem, a temporização, frequência, escalonamento, tipo e tamanho dos pacotes, de forma a

emular várias aplicações. Também se pode controlar estatísticas de qualidade e o caminho para serem monitorados.

3.3. Medidas Relativas

De acordo com o tipo e a utilização das grandezas, podem ser definidos dois tipos de medições: medições relativas à utilização de recursos de rede e medições de tráfego de acordo com desempenho. Para cada tipo, deve ser especificado um conjunto de pontos de medição de forma a criar segmentos de rede específicos para os propósitos de medição.

3.3.1. Medições relativas à utilização de recursos

Este tipo de medição utiliza um conjunto de métricas compreendendo as diferentes utilizações para cada recurso da rede, como, por exemplo, a capacidade de processamento e memória dos roteadores, pelos diferentes tipos de tráfego.

3.3.2. Medições de tráfego relativas ao desempenho

Este tipo de medição é o foco deste trabalho. O principal componente da gerência do desempenho é a coleta contínua, e em tempo real, da qualidade e da condição da rede e de seus vários elementos, para assegurar que seja sustentado um serviço ininterrupto com alto desempenho. Isto requer o uso de medições ativas para coletar informações sobre o estado operacional da rede e seu desempenho. Dentre as métricas mais comuns estão latência, *jitter*, perda de pacotes e vazão. Os trabalhos mais significativos no sentido de estabelecer um conjunto de métricas úteis, claras e de fácil entendimento e utilização por parte de usuários e operadores, são os desenvolvidos pelo ITU-T [ITU08] e pelo IETF [IETF08], RFCs e *drafts* do grupo de trabalho *IP Performance Metrics - IPPM* [IPPM08].

3.3.2.1. Medições Unidirecionais

Medições unidirecionais são aquelas que medem o desempenho da rede em apenas um sentido. Entre elas estão o atraso, o *jitter* e a perda de pacotes. Essas medições são necessárias para caracterizar o estado dos caminhos entre os *hosts*, pois as rotas podem ser assimétricas, ou seja, os caminhos de ida e volta podem ser diferentes e o enfileiramento dos pacotes pode sofrer variações.

Erros e incertezas nos métodos de medição unidirecionais são fortemente influenciados pelas incertezas dos relógios de origem e destino. Os relógios devem estar sincronizados, pois

qualquer erro de sincronização irá contribuir no erro da medida do retardo e, conseqüentemente, também da perda. A resolução de ambos os relógios também contribui para o erro, pois seu valor será adicionado às medidas obtidas. Outras fontes de erros e incertezas para perda são: o limite de perda de pacotes que está relacionado à sincronização e a limitação de recursos da interface de rede ou do software da máquina de recepção, que podem causar o descarte de pacotes, fazendo com que o pacote de teste seja considerado como perdido mesmo tendo alcançado seu destino.

3.3.2.2. Medições de Ida e Volta (*Round Trip Time Measurement*)

Muitos caminhos na Internet são assimétricos, ou seja, a seqüência de roteadores atravessados por um pacote transmitido de uma origem para um destino é diferente da seqüência atravessada no retorno deste mesmo destino para esta mesma fonte. Na presença de caminhos assimétricos, medições tradicionais de *round-trip* indicam o desempenho de dois caminhos diferentes em conjunto.

Uma das vantagens da utilização de medidas de ida e volta é a facilidade para a interpretação dos resultados, ou seja, as medidas de ida e volta (RTT) são mais fáceis de serem obtidas e não necessitam de nenhum equipamento externo para sincronização das máquinas de recepção.

Portanto, é interessante considerar o que pode estar sendo perdido quando não são feitas medidas unidirecionais. A maioria dos caminhos na Internet é assimétrica. Entretanto, para a maior parte das aplicações, especialmente aquelas que utilizam o TCP como protocolo de transporte, não faz diferença se o retardo é simétrico ou assimétrico, sendo o tempo total de ida e volta o que realmente influencia no desempenho da aplicação. Entre estas aplicações estão incluídas a *Web* [WIKI08]. Uma exceção importante é voz sobre IP, onde enlaces que possuem retardo altamente assimétrico podem tornar possível o tráfego de voz em uma direção, e impossível na outra. Presumivelmente outras aplicações de multimídia de tempo real que não possuem uma “realimentação”, como vídeo, também são afetadas pelo retardo assimétrico.

3.4. Métricas mais comuns segundo IEEE e ITU-T

Existem várias propriedades (quantidades) relacionadas com o desempenho de uma rede. Quando uma destas quantidades é especificada, ou seja, quando se refere a uma propriedade de um componente, esta é denominada métrica. Da mesma forma, valores quantificados de uma

métrica são denominados medidas. Quando uma métrica é quantificada, sua medida deve ser apresentada em termos de unidades padronizadas.

Desta forma, esta seção descreve as métricas mais comumente usadas para descrever o desempenho de redes. São métricas em geral derivadas de definições do IEEE [IEEE08] e do ITU-T [ITUT08], para redes telefônicas e para redes de dados.

3.4.1. Latência

Latência para redes de computadores, se refere a quanto tempo um pacote leva para ir de um ponto para outro.

A latência de rede possui os seguintes componentes:

- Tempo de transporte: O tempo que o pacote leva para ser transmitido através dos enlaces físicos que compõem o caminho através da rede.
- Tempo de enfileiramento (*queuing*): O tempo de espera nas filas internas dos roteadores, que varia de acordo com o algoritmo de escalonamento e com políticas de *Quality of Service* - QoS e policiamento quando adotadas.
- Tempo de envio/transferência: O tempo necessário para o roteador receber o pacote na sua interface de entrada, processá-lo e, caso haja necessidade, fazer o policiamento e condicionamento do tráfego e encaminhar para a interface de saída.
- Tempo de resposta do servidor: O tempo requerido pelo servidor para processar um pacote que entra e gerar um pacote de resposta.

A medida da latência da rede pode ser feita através de cada componente separadamente: o retardo de envio do pacote, ou seja, o tempo de transporte somado ao tempo de enfileiramento e do envio/transferência; o retardo do servidor, ou seja, seu tempo de resposta; e o retardo de retorno, que freqüentemente não é o mesmo de ida, uma vez que os caminhos podem ser assimétricos. Medir os retardos de envio e retorno requer medições tanto do lado do cliente como do servidor, usando técnicas de retardo unidirecional. Outra forma mais direta de obter a latência da rede é medir a soma de todos os componentes para produzir um único valor.

Valores típicos de latência variam de centenas de microssegundos em uma rede local, a centenas de milissegundos, como é o caso de enlaces via satélite.

A latência não é uma medida fixa, variando de acordo com as condições da rede. A carga do servidor, o congestionamento da rede e mudanças de roteamento influenciam o seu valor final. Se o servidor está pouco carregado irá responder rapidamente, porém servidores ocupados serão mais lentos, o que irá aumentar o tempo de resposta do servidor. Da mesma forma, se não

há congestionamento no caminho do pacote, o tempo de espera nas filas internas dos roteadores será mínimo, reduzindo o tempo de enfileiramento. Entretanto, se o caminho está congestionado, este tempo pode aumentar muito, chegando a algumas situações de descarte de pacotes, contribuindo para o aumento da latência. Mudanças de roteamento também influenciam o valor desta métrica.

Para detectar variações na latência é comum produzir gráficos diários mostrando a latência média para pequenos intervalos. Esses gráficos frequentemente mostram variações diurnas, com a latência aumentando durante os períodos de maior utilização do dia. Aumentos ou reduções repentinas, causadas por mudanças de rotas, ataques à segurança da rede, ou simplesmente rajadas de tráfego, também podem ser mostradas.

3.4.2. Perda de Pacotes

A perda de pacotes na rede é a fração dos pacotes perdidos no trânsito entre origem e destino durante um intervalo de tempo especificado, expressa como uma porcentagem dos pacotes enviados para o destino durante este intervalo. Esta pode variar conforme o congestionamento do caminho até o valor máximo definido para o serviço ou, caso não haja nenhuma definição, até que seja impossível recuperar a informação enviada. Altas taxas de perda podem tornar a rede inviável.

Um segundo ponto particularmente importante, é o fato de que o TCP [TCP81] utiliza a detecção de pacotes perdidos como um meio de perceber o congestionamento da rede. Assim é esperada uma perda ocasional de pacotes em fluxos TCP. Deste modo, uma perda moderada de pacotes, não é por si só uma indicação de falha de rede. Alguns serviços podem tolerar a perda de alguns pacotes e estes, na maioria das vezes, são reenviados pelo TCP que usa a perda de pacote como um sinal indicando a necessidade de enviar os dados a uma taxa mais baixa. Assim, muitos serviços irão continuar a operar efetivamente, mesmo em face de alguma perda de pacotes.

3.4.3. Vazão

Vazão é a taxa com que os dados são enviados através da rede. Na maioria das vezes esta métrica se refere à taxa total de transferência dos dados para todo o tráfego sendo transportado, porém também pode ser útil medir a vazão a uma granularidade mais fina, como, por exemplo, para transações multimídia para voz sobre IP ou para destinos específicos e etc.

A vazão é medida continuamente pela contagem de *bytes* transportados durante um intervalo de tempo especificado, sendo que a escolha deste intervalo deve ser feita

cuidadosamente, pois intervalos longos tendem a incorporar as rajadas, possíveis apenas por breves períodos, ao valor medido. Por outro lado, intervalos pequenos implicam numa taxa de coleta de dados maior, e podem medir as rajadas de forma exagerada.

3.4.4. Capacidade

A capacidade do enlace é a vazão máxima que um caminho pode prover para uma aplicação quando não há outro tráfego competindo pelos recursos (*cross traffic*). A medição da capacidade é crucial para detecção de problemas, calibração e gerenciamento de caminhos.

A capacidade de um caminho é determinada pelo enlace de menor capacidade (*narrow link*). A capacidade de um enlace é determinada pelo seu meio físico e pela tecnologia de transmissão utilizada. Por exemplo, a capacidade teórica de enlaces utilizando fibras óticas é da ordem de Terabits por segundo. Entretanto, as tecnologias de transmissão óptica como CWDM [G.694.2] e DWDM [G.694.1] limitam a vazão máxima a *Gigabits* por segundo.

3.4.5. Utilização do Enlace

Os enlaces possuem uma taxa máxima de transmissão de dados, conhecida como taxa de acesso. A utilização do enlace durante um intervalo de tempo específico é simplesmente a vazão deste enlace durante o intervalo determinado sem perda de pacotes, expressa como uma porcentagem da sua taxa de acesso.

3.4.6. Banda disponível

Banda disponível é a vazão máxima que um caminho pode prover para uma aplicação, dada a atual carga de tráfego do caminho, ou seja, a banda disponível sob uma determinada utilização do enlace ou do caminho. A banda disponível de um caminho é determinada pelo enlace com a mínima capacidade sem utilização.

3.5. Métricas mais comuns segundo IPPM – IETF

O IETF, através do grupo de trabalho *IP Performance Metrics* [IPPM08] definiu e recomenda algumas métricas para avaliação do desempenho da rede [RFC2672]. Essas métricas se destinam a definir as principais características que influenciam o desempenho da rede e das aplicações suportadas. Algumas aplicações são sensíveis a uma ou mais grandezas, tendo o seu desempenho degradado se as propriedades da rede entre as máquinas estiverem acima de valores limites. Por exemplo, retardo e/ou perda de pacotes excessiva torna difícil suportar certas

aplicações de tempo real. O valor preciso do limite para ser considerado excesso depende da aplicação. Quanto pior estejam as características da rede, se torna mais difícil para os protocolos de camada de transporte sustentar altas taxas. Assim, é de extrema importância definir e mensurar tais características.

Todas as métricas propostas pelo IPPM têm seu foco na rede e foram desenvolvidas considerando um pacote genérico usado para obtenção das medidas. O valor da métrica pode depender das propriedades do pacote realmente utilizado. Também devem ser considerados os caminhos atravessados pelos pacotes de teste, a calibração dos erros, e o limite definido para o retardo.

3.5.1. Conectividade

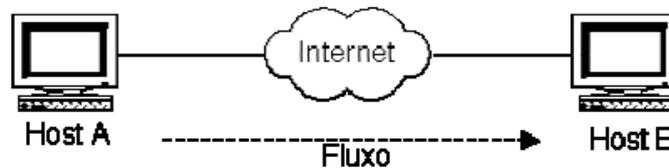
Conectividade é a junção das diversas máquinas com várias redes, permitindo a troca de dados na Internet. O IPPM definiu várias métricas para conectividade, algumas das quais servem principalmente como parte integrante de outras métricas. Estão relacionadas a seguir as principais métricas para medir conectividade:

- Conectividade instantânea unidirecional: Duas máquinas origem e destino têm conectividade instantânea unidirecional em um tempo T , se um pacote transmitido pela origem no tempo T , chegar ao destino.
- Conectividade instantânea bidirecional: Duas máquinas origem e destino têm conectividade instantânea bidirecional, se a origem tem conectividade instantânea unidirecional para o destino, e o destino tem conectividade instantânea unidirecional para a origem, ou seja, origem e destino estão totalmente conectados no tempo T .
- Conectividade unidirecional: A origem tem conectividade unidirecional para o destino durante o intervalo $[T, T+dt]$, se para qualquer T pertencente a $[T, T+dt]$ a origem tem conectividade instantânea unidirecional para o destino.
- Conectividade bidirecional: Origem e destino têm conectividade bidirecional entre si durante um intervalo $[T, T+dT]$, se a origem tem conectividade unidirecional para o destino durante o intervalo, e se o destino tem conectividade unidirecional para a origem durante o intervalo.
- Conectividade bidirecional temporal: A origem envia um pacote para o destino que encaminha uma resposta.

3.5.2. Retardo Unidirecional

Para melhor entendimento dessa medição, considere um fluxo no sentido do host A para o host B como mostrado na Figura 1. A perda unidirecional ocorre quando o pacote enviado por A não chega a B. Quando ocorre fragmentação de um pacote, a perda é caracterizada se pelo menos um dos fragmentos é perdido.

Figura 1: Sentido do fluxo. [RSBG08]



O retardo unidirecional é definido como sendo o tempo que o pacote demora a sair da origem e chegar ao destino. Para obtenção dessa métrica, tanto origem como destino precisam ter os relógios sincronizados com uma boa precisão. Caso haja fragmentação do pacote analisado, o atraso é dado em função do último fragmento recebido, caso todos tenham sido recebidos.

A variação do atraso em um sentido pode ser definida sobre dois pacotes quaisquer contidos no fluxo $A \Rightarrow B$.

A determinação do retardo e da perda de pacotes unidirecionais são úteis em especial pelas indicações que provêm sobre a rede:

- O valor mínimo do retardo unidirecional provê uma indicação do retardo devido apenas ao tempo de propagação e transmissão.
- O valor mínimo do retardo unidirecional provê uma indicação do retardo que é experimentado quando o caminho estiver com baixa utilização.
- Valores do retardo unidirecional acima do mínimo indicam congestionamento presente no caminho.

A sensibilidade de aplicações de tempo real e de determinados protocolos da camada de transporte à perda de pacotes, se torna especialmente importante quando produtos que consomem muita banda e possuem alto retardo, devem ser suportados.

3.5.3. Retardo de Ida e Volta (*Round Trip Time*)

Assim como as medidas unidirecionais, a medição do retardo de ida e volta pode trazer importantes indicações sobre o desempenho da rede. Como dito anteriormente, a medição do retardo de ida e volta é de fácil implementação e utilização. Na maioria das vezes não é preciso nenhuma instalação ou configuração especial no destino. Mesmo as ferramentas utilizadas pela

origem são bem conhecidas, o que facilita a realização das medições e a interpretação dos seus resultados.

O retardo de ida e volta é o intervalo de tempo dT entre: o envio pela origem do primeiro bit de um pacote em um tempo T , o recebimento e o retorno imediato do mesmo pacote pelo destino do último bit em um tempo $T+dT$. O retardo é indefinido, caso o pacote não seja recebido pelo destino ou o pacote de resposta não seja recebido pela origem.

Da mesma forma que para o retardo unidirecional, pacotes corrompidos ou que não podem ser remontados são considerados como não recebidos. Pacotes duplicados são considerados recebidos, e o primeiro a chegar à origem é utilizado para determinar o retardo de ida e volta [RFC2681].

3.5.4. Variação do Retardo (*Jitter*)

Essa métrica é baseada na diferença do retardo unidirecional de pacotes selecionados. A diferença dos retardos é chamada variação do retardo de pacotes IP (*IPDV – IP Packet Delay Variation*).

Um importante uso da variação do retardo é a determinação da dinâmica de filas em uma rede ou um roteador, uma vez que mudanças na variação do retardo podem estar relacionadas a mudanças no tamanho das filas em um dado enlace ou combinação de enlaces.

Esse tipo de métrica pode ser utilizado mesmo com diferenças e variações nos relógios entre duas máquinas. Isso permite o seu uso mesmo se as duas máquinas que suportam as medições não estiverem sincronizadas. Se como uma primeira aproximação, o erro que afeta a primeira medida do retardo unidirecional for o mesmo que afeta a segunda medida, eles serão cancelados durante o cálculo do IPDV.

3.6. Comparativo entre Organismos de Padronização

O *Internet Engineering Task Force* - IETF, tem sido por muitos anos o principal responsável pela preparação de padrões para a Internet, isto inclui o IP e muitos dos serviços suportados. O grupo de trabalho do IETF que trata de métricas de desempenho para o protocolo IP (*IP Performance Metrics*), tem produzido documentos que especificam essas métricas e propõem metodologias para o seu cálculo. Existem algumas ferramentas de medição ativa que foram criadas com base nesses documentos. No entanto, os métodos propostos pelo grupo de trabalho IPPM para o cálculo de métricas unidirecionais, apenas caracterizam o estado de um

caminho da *Internet* e não avaliam o desempenho do serviço oferecido aos pacotes de uma determinada aplicação.

As métricas do IEEE e do ITU-T possui como meta o estabelecimento de padrões para formatos de computadores e dispositivos visualizando a área de telecomunicações. Igualmente importantes estes organismos estão dedicados a aumentar os mecanismos de medidas da *Internet* através do desenvolvimento de novas ferramentas e com a criação de novos padrões.

As métricas tradicionalmente utilizadas, baseadas em padrões do IEEE [IEEE08] e do ITU-T [ITUT08] e as métricas desenvolvidas pelo IETF [IETF08] para redes IP, muitas vezes lidam com a mesma grandeza ainda que com enfoques diferentes. Isto faz com que freqüentemente haja uma falta de entendimento sobre o que está sendo medido e o significado dos resultados das medições. Veja no quadro 1, as medidas relacionadas ao atraso unidirecional (OWD) através dos organismos apresentados neste capítulo.

Quadro 1 : Atraso Unidirecional

Cálculo de Atraso Unidirecional (OWD)	
IETF	Dado um conjunto de pelo menos dois pacotes entre A e B, é a diferença no OWD de um par selecionado de pacotes no conjunto.
ITU-T	Diferença entre o percentil 99,9 do OWD e um atraso referência.

No quadro 1, se considerarmos o atraso referência como sendo o atraso mínimo do caminho, se for escolhido um dos pacotes anteriores da amostra de pacotes, a definição do ITU-T e do IETF coincidem.

3.7. Conclusão

A realização de medições fim-fim se torna útil para a obtenção de medidas em uma rede amplamente distribuída. Esse tipo de medida, geralmente provê informações sobre a aplicação como um todo, incluindo o desempenho de clientes e servidores, como também da rede entre eles. Portanto, todas as métricas e metodologias de medição descritas neste capítulo demonstram a importância da existência de métricas bem definidas, identificando os parâmetros de qualidade e desempenho de redes, pois atualmente existem várias ferramentas de medições ativas que foram criadas com base nesses documentos.

Capítulo 4

Sincronismo dos Relógios em Rede

4.1. Introdução

Em redes que requerem alta velocidade, fatores como a diferença do tempo de transmissão se tornam essenciais para os vários computadores que trocam informações a cada segundo ou, até mesmo, na ordem de microssegundos. É justamente nesse ambiente de partilha de informação que a necessidade de uma referência de tempo comum, precisa e confiável, se torna imprescindível.

Da mesma forma, no âmbito dos sistemas computacionais, vários programas e aplicações dependem da sincronização correta dos relógios, principalmente em aplicações que dizem respeito a estudos de medições em redes. A falta de sincronização pode levar a falha na execução correta das medições, além da perda dos dados a serem medidos. No entanto, sincronizar relógios em rede é um problema difícil porque o tempo é um alvo móvel.

Para o cálculo das métricas que são função do tempo, como o *delay* (atraso) e o *jitter* (variação do atraso), é necessário prestar atenção na qualidade dos mecanismos de operação do relógio das máquinas usadas para capturar o tráfego. Em princípio, dado que os atrasos da rede são imprevisíveis, supõe-se que é absolutamente necessário garantir que os relógios das máquinas usadas para coletas e/ou geração de tráfego estejam sincronizados. Para analisar até onde vai essa necessidade, neste capítulo estão descritos, de forma resumida, os mecanismos que são usados pelas máquinas para computar o tempo, e os mecanismos usados para sincronizar os relógios de computadores em rede.

4.2. Alguns Conceitos em Medição na Sincronização de Relógios em rede

Um período solar tende a divergir em relação ao período atômico. As voltas da terra em torno do sol não gastam sempre o mesmo tempo atômico, ou seja, um dia solar não contém a mesma quantidade de segundos de um relógio físico. Em geral, isso ocorre devido a vários fenômenos astronômicos.

O *Coordinated Universal Time* - UTC é um padrão internacional baseado no período atômico, mas disciplinado com o período solar. Para fazer a correção do padrão UTC, é utilizado um fator de ajuste de 1 segundo, chamado *leap second*, o qual é inserido ou removido a cada 18 meses aproximadamente. Sinais UTC são sincronizados e difundidos em *broadcast* por estações de rádio na terra e através de satélites, para quase todas as partes do mundo.

Um relógio físico consiste de três mecanismos: um oscilador, um contador de oscilações que registra continuamente o número de ciclos do oscilador, e um mecanismo para mostrar, gravar ou exibir o resultado. Um oscilador é um mecanismo que repete eventos a uma taxa constante, cujo erro é normalmente expresso em PPM (partes-por-milhão). Este é também chamado de frequência. O contador converte o número de ciclos contados no oscilador em unidades de tempo conhecidas, tais como hora, minuto e segundo. Cada resultado, ou seja, cada valor do contador é uma estampa do tempo (*timestamp*).

Dois conceitos são relacionados à sincronização de relógios físicos e manutenção de tempo.

- Exatidão – é o grau de conformidade de uma medida de tempo em relação ao valor verdadeiro ou referência padrão, neste caso conhecida como a “hora certa” (UTC).
- Resolução ou precisão – caracteriza-se por ser a unidade da menor estampa de tempo (*timestamp*) que um relógio pode gerar, seja minuto, segundo, ou unidades menores.

Cada relógio físico tem algumas características que influenciam bastante a qualidade do tempo e uma característica importante é a resolução do tempo fornecido. Em sua maioria, na placa mãe dos computadores encontra-se um relógio físico (*hardware*) barato, com uma resolução na faixa de 10ms (100Hz), onde, tipicamente, este só é acessado em duas ocasiões: para leitura quando o sistema é ligado (*boot*) e para gravação quando o sistema é desligado. Paralelamente, os sistemas operacionais provêm um relógio, chamado de relógio do sistema, baseado em código (*software*) e na capacidade do processador, para fornecer a hora às aplicações. A resolução do relógio do sistema é na faixa de 1s, variando em função do sistema operacional usado.

Outra característica importante é a precisão do relógio que define a granularidade efetiva da informação de tempo, mesmo que se tenha uma resolução mais alta. Assim, um relógio que tem uma resolução de tempo de 1s pode estar fornecendo a hora com uma precisão de 10ms. A precisão varia bastante entre sistemas operacionais, tipicamente de 10ms (*Windows*) a poucos microssegundos, e em alguns sistemas operacionais esta é ajustável (*FreeBSD*, *Linux*), com recompilação do kernel.

Existem alguns relógios de sistema (sistemas *Unix*), também chamados de *nanokernels*, que permitem a um relógio trabalhar com a resolução e precisão de poucos nanosegundos.

Comparação entre relógios

Para ser feita a comparação de relógios físicos, existem 3 métricas na literatura [MILLS92] que podem ser usadas para avaliar a qualidade de um relógio em relação a um relógio de referência:

- a) Diferença entre um relógio e o de referência em um dado instante. Baseado nessa métrica pode-se ajustar a fase (sincronização) do relógio.
- b) Desvio de um relógio em relação ao relógio de referência, que é o aumento ou diminuição da diferença entre dois relógios em um intervalo de tempo. Esta métrica está diretamente relacionada ao erro em frequência.
- c) Tendência, que é a variação do desvio em relação a um relógio de referência, em função do tempo.

Baseado nisso um bom mecanismo de sincronização de relógios tem 2 objetivos principais: ajuste de fase e ajuste de frequência. Para tal, é necessário não apenas “ajustar” o relógio local, mas alterar o seu comportamento.

4.3. Relógios de Alta Precisão

É possível utilizar um relógio físico de alta precisão para corrigir o comportamento do relógio de sistema de um computador, dentre eles:

- O GPS é baseado em uma rede de satélites mantida pelo governo americano. Este sistema usa uma antena externa que pode se conectar diretamente a uma placa receptora instalada em um computador (esquema com maior precisão) ou então se conectar a um receptor que irá transmitir a informação ao computador através da porta serial (menor precisão). As diversas implementações de GPS têm precisão entre dezenas de microssegundos até poucos nanosegundos.

- O Relógio Atômico que é baseado em um átomo oscilador de alta precisão. Os mais comuns são os baseados em átomos de Césio, Rubídio ou Hidrogênio.
- O Receptor de avisos de hora por ondas de rádio, que interpreta sinais de hora provenientes de estações de disseminação. A interconexão com um computador é feito através da interface serial (menor precisão) ou de uma placa de som (maior precisão).

4.4. Sincronização de Relógios em rede

Cada computador possui seu próprio relógio interno, e dois processos executados em computadores diferentes através de uma rede distribuída tendem a associar indicações de tempo diferentes, já que cada relógio local apresenta uma taxa de *drift* (desvio) em relação aos demais relógios.

Mesmo que os relógios de todos os computadores fossem ajustados no mesmo horário, com o passar do tempo estes iriam apresentar variações destas taxas entre si. No entanto, é possível saber o limite de sincronização externa quando os relógios dos processos estão sincronizados com uma autoridade externa, e o limite de sincronização interna quando os relógios dos processos estão sincronizados entre eles dentro de um determinado limite de tempo [COUL08].

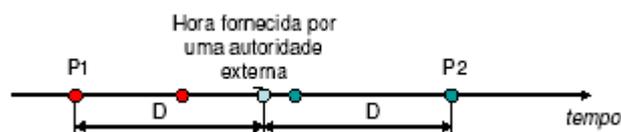
Sincronização Externa

Utilizada quando os relógios dos processos estão sincronizados com um relógio externo.

A figura 2 representa um exemplo de sincronização externa.

- $|S(t) - C_i(t)| \leq D$ para $i=1,2,\dots,N$
- t: tempo físico/real.
- S: tempo dado por um servidor UTC.
- C_i : clock do i-ésimo computador.
- D: limite de sincronização.

Figura 2: Hora Fornecida por uma Autoridade Externa. [AUT08]



Sincronização Interna

Utilizada quando é necessário somente controlar o tempo de execução dos processos internos.

- $|C_i(t) - C_j(t)| \leq D$ para $i, j=1, 2, \dots, N$
- t: tempo físico/real.
- $C_{i,j}$: clock do i-ésimo computador.
- D: limite de sincronização.

Se um conjunto de processos está sincronizado externamente dentro do limite D, então estes processos estão sincronizados internamente dentro do limite 2D [COUL08]. Outra maneira de se expressar isto, é que os relógios C_i são precisos dentro do limite D.

4.5. Algoritmos de Sincronização Interna e Externa

Nos itens a seguir são descritos alguns algoritmos que utilizam a sincronização interna e externa, baseados em trocas periódicas de mensagens [COUL08].

4.5.1. Algoritmo de Cristian

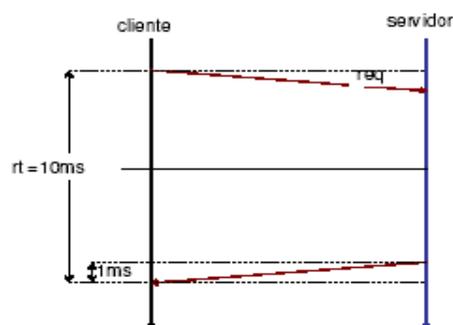
Este algoritmo é de sincronização externa, e sugere um servidor de tempo conectado a um dispositivo que recebe sinais de uma fonte UTC [COUL08]. Deste modo, ao receber uma mensagem de requisição, o processo deste servidor S fornece o tempo de acordo com seu relógio. Veja através da figura 3 um exemplo de funcionamento do algoritmo de Cristian.

Figura 3: Modo de Funcionamento do algoritmo de Cristian. [AUT08]



Se o tempo de transmissão mínimo for conhecido, pode-se obter o erro máximo de precisão. A figura 4 demonstra como é feita a troca de mensagens deste algoritmo.

Figura 4: Troca de mensagens do algoritmo de Cristian. [AUT08]



- $T_{Min} = 1ms$, e que o $RTT = 10ms$, então $RTT/2 - T_{min} = 10 / 2 - 1 = 4ms$. Este seria então o erro máximo de precisão.

4.5.2. Algoritmo Berkeley:

Este é um algoritmo de sincronização interna desenvolvido para um conjunto de computadores executando Unix Berkeley [COUL08]. Um computador coordenador é escolhido para atuar como “mestre”, fazendo consulta seqüencial e periódica a outros computadores “escravos”, enviando os valores de seus relógios. O mestre faz uma estimativa dos tempos locais destes relógios, observando o RTT (*Round Trip Time*) e fazendo a média dos valores obtidos, logo após envia o valor de ajuste aos escravos. Se o mestre falhar, outro computador pode assumir. A média estimativa elimina a tendência causada por certos relógios que andam mais rápidos ou mais lentos que outros.

4.5.3. NTP - *Network Time Protocol*

Tanto o método de Cristian como o de Berkeley se destinam principalmente ao uso na intranet. Por outro lado, o protocolo NTP define uma arquitetura para um serviço de tempo, distribuindo informações para redes amplamente distribuídas como a *Internet*.

O NTP [NTP08] faz os ajustes de fase e frequência através de um disciplinador de relógio, com o objetivo de melhorar a qualidade da hora fornecida pelo relógio de sistema em qualquer instante. Ambos são alimentados por dados obtidos através de requisições a relógios de referência na Internet, e por dados obtidos do próprio relógio de sistema local para efeito de comparação. Tendo-se uma boa conexão de rede até um relógio de referência, o NTP pode ser configurado para garantir um erro inferior na ordem de microsegundos.

4.5.3.1. Arquitetura do protocolo NTP

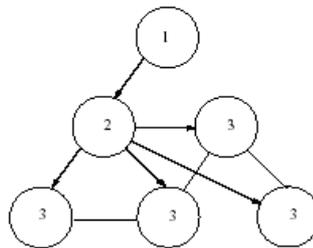
O protocolo NTP utiliza como meio de comunicação redes baseadas no protocolo IP, implementando um modelo de sincronização hierárquico distribuído, apresentando conceito de *stratum* (estrato). Os servidores de *stratum* 1 são máquinas ligadas diretamente a dispositivos conhecidos como "relógios de referência" (ou servidores *stratum* 0) de elevada precisão. Estes dispositivos podem ser relógios atômicos, receptores GPS, ou receptores de rádio. Este protocolo utiliza a camada de transporte UDP e a comunicação entre cliente e servidor é feita através da porta 123.

Qualquer servidor NTP que tenha como referência de tempo um servidor *stratum* 1 passa a ser um *stratum* 2, qualquer Servidor NTP que tenha como referência de tempo um servidor *stratum* 2 passa a ser um *stratum* 3, e assim por diante. Na prática, os clientes/servidores NTP

são ligados a vários outros servidores com o *stratum* mais baixo possível, e com o menor caminho de rede entre eles.

A Figura 5 apresenta de forma esquemática uma sub-rede NTP, onde os números em cada nó representam o valor do *stratum* e cada servidor na sub-rede de sincronização é configurado com outro servidor denominado par (*peer*) de sincronização. As linhas em destaque representam uma sincronização ativa e a direção do fluxo da informação de tempo. As linhas mais finas representam caminhos dos quais a informação de tempo é trocada entre os pares, mas não necessariamente utilizada para sincronização dos relógios locais.

Figura 5: Estrutura Hierárquica do NTP. [WPER08]



4.5.3.2. Tipos de associação

As relações entre os diferentes servidores NTP são normalmente chamadas de associações. Elas podem ser:

- Permanentes: criadas por uma configuração ou comando e mantidas sempre.
- Priorizáveis (*preemptables*): criadas por uma configuração ou comando, podendo ser desfeitas no caso de haver um servidor melhor, ou depois de um certo tempo.
- Efêmeras ou transitórias: criadas por solicitação de outro dispositivo NTP, podendo ser desfeitas em caso de erro ou depois de um certo tempo.

Os servidores NTP são sincronizados entre si de três (3) maneiras:

- Cliente - Servidor: considerada uma associação permanente e a forma mais comum de configuração, onde um dispositivo faz o papel de cliente, solicitando informações sobre o tempo a um servidor, e o servidor responde à solicitação do cliente com informações sobre o tempo.
- Simétrica: dois ou mais dispositivos NTP podem ser configurados como pares (*peers*), de forma que possam tanto buscar o tempo, quanto fornecê-lo, garantindo redundância mútua. Essa configuração faz sentido para dispositivos no mesmo estrato, configurados também como clientes de um ou mais servidores.

- *Broadcast* ou *Multicast*: O NTP pode fazer uso de pacotes do tipo *broadcast* ou *multicast* para enviar ou receber mensagens de tempo. Esse tipo de configuração pode ser vantajosa no caso de redes locais com poucos servidores alimentando uma grande quantidade de clientes.

4.5.3.3. Métricas relativas à Qualidade da Sincronização

São as métricas que caracterizam a qualidade da sincronização obtida e mantida por uma rede NTP [NTP08]. O NTP calcula as métricas *clock offset*, RTT, *jitter* e dispersão.

- *Clock Offset*: O *clock offset*, ou simplesmente *offset*, é o valor do ajuste necessário ao relógio local para colocá-lo em fase com o relógio de referência, ou seja, é a diferença entre esses relógios. É o indicador mais importante da qualidade da sincronização. Quanto menor o *offset*, melhor é a sincronização.
- RTT - Atraso para cada associação (par) e a raiz. A métrica de atraso ou distância entre dois nós da rede é o RTT. O protocolo NTP calcula o atraso para cada associação e a distância para a raiz, que representa o valor acumulado do RTT para a origem da referência de tempo primária na sub-rede de sincronização. O NTP assume que o caminho de rede entre *hosts* é simétrico.
- *Jitter*: Para os pares de strato, esse valor é calculado pelo algoritmo *clock filter* e, em seguida, utilizado pelo algoritmo *clustering* como uma métrica de qualidade que representa a variação nas amostras de *offset* para cada associação. Essa variação ocorre, em geral, devido a características na infra-estrutura da rede, como variação do atraso, caminhos congestionados ou assimétricos. Para o sistema, esse valor representa a melhor estimativa de erro na computação do *offset*.
- Dispersão para a raiz e para a associação - O protocolo NTP calcula também a dispersão para a raiz e para o par (associação) A dispersão indica o erro máximo em relação à fonte de referência (raiz ou par);

4.5.3.4. Algoritmos de Sincronização

Funções associadas:

1. Obter, à partir de diversas amostras, informações de tempo de um determinado servidor, como o deslocamento (diferença de tempo entre dois relógios), dispersão (desvio ou erro estimado nas leituras do relógio), e *jitter* (variação das amostras do *offset*).
2. Discernir, dentre um conjunto de servidores, quais fornecem o tempo correto.

3. Escolher, dentre os servidores que fornecem o tempo correto, qual é a melhor referência.
4. Disciplinar o relógio local, descobrindo seus principais parâmetros de funcionamento, como precisão, e escorregamento (instabilidade na frequência do oscilador), ajustando-o de forma contínua e gradual, mesmo na ausência temporária de referências de tempo confiáveis, para que tenha a melhor exatidão possível.

Para executar essas funções, o NTP utiliza os diferentes tipos de algoritmos descritos a seguir.

Algoritmo de Filtro de Relógio

Através da troca de mensagens, o NTP consegue as informações de atraso e deslocamento do relógio local para um servidor. Essa troca de mensagens se repete, periodicamente com um período dinamicamente controlado pelo NTP. Não há apenas um valor de atraso e um de deslocamento para cada servidor, mas um conjunto deles, provenientes das diversas trocas de mensagens. A partir dessa lista de valores, este algoritmo calcula o valor de atraso (*delay*), de deslocamento (*offset*) e de variação (*jitter*).

Cada amostra da mensagem, é composta de 4 valores: atraso, deslocamento, dispersão e estampa de tempo (*timestamp*). A estampa de tempo indica quando a amostra chegou. A dispersão é o erro estimado do relógio do servidor remoto.

A lista com os valores é ordenada em função do atraso, ou seja, as amostras com menor atraso são consideradas melhores e, deste modo, é formada uma lista com as amostras mais recentes e ordenadas em função do atraso. Da primeira entrada dessa lista são retiradas as variáveis atraso e deslocamento (para cada par cliente - servidor há uma variável de cada tipo).

A dispersão e a variação são calculadas levando-se em conta todos os valores da lista.

Algoritmo de Seleção dos Relógios

Com o cálculo dos principais parâmetros referentes a cada servidor interligado, este algoritmo tenta descobrir quais deles são confiáveis e quais não. Para a seleção dos relógios confiáveis, o NTP considera como verdadeiro o deslocamento que se encontra dentro de um determinado intervalo de confiança. Esse intervalo é calculado, para cada associação com um servidor, da seguinte forma:

- intervalo de confiança = (deslocamento/2) + dispersão.

Os servidores cujos valores de deslocamento ficam fora do intervalo de confiança, são considerados como relógios falsos.

Algoritmo de Agrupamento

Este algoritmo trabalha com os servidores que são relógios verdadeiros, utilizando técnicas estatísticas com o objetivo de selecionar os melhores dentre eles. Os critérios de seleção utilizados são:

- Valor do estrato (*stratum*)
- A distância para a raiz
- A variação do *offset* (*jitter*)

Neste processo alguns servidores são descartados, sendo chamados de relógios afastados (*outliers*). Os que permanecem são chamados de relógios sobreviventes (*survivors*). O melhor dos relógios sobreviventes é considerado como par do sistema (*system peer*).

Algoritmo de Combinação de Relógios

Se o algoritmo de agrupamento encontrar apenas um sobrevivente, ele será o par do sistema, e é utilizado como referência para ajustar o relógio local. Se um servidor for configurado como tendo preferência sobre os demais e estiver dentre os sobreviventes, ele será considerado como par do sistema e mesmo que existam outros sobreviventes, estes serão ignorados. Nestes casos, este algoritmo não é utilizado.

Para os demais casos, o algoritmo de combinação de relógios calcula uma média ponderada dos deslocamentos dos relógios, com o objetivo de aumentar a exatidão.

Algoritmo de Disciplina do Relógio Local

O processo de disciplina controla a fase e a frequência do relógio do sistema. Ele é baseado na combinação de duas filosofias de controle bastante diferentes entre si: *Phase Locked Loop* (PLL) e *Frequency Locked Loop* (FLL)

- O controle baseado em fase é melhor para as ocasiões onde há uma grande variação (*jitter*). Essa abordagem procura minimizar o erro no tempo, controlando indiretamente a frequência.

- O controle baseado em frequência é melhor quando há instabilidades na frequência (variações mais lentas que o *jitter*). Esta abordagem controla diretamente a frequência, e indiretamente o erro no tempo.

O NTP disciplina o relógio local de forma contínua e gradual, mesmo em períodos onde não é possível consultar servidores de tempo. As características do relógio local são medidas, e se tornam conhecidas do NTP.

4.6. Conclusão

A falta de sincronização em uma rede pode gerar falhas na execução correta das tarefas do sistema, bem como perda de dados, problemas de segurança e de confiabilidade. Além disso, a qualidade da sincronização depende da exatidão e da resolução dos relógios das fontes de referência, da troca de mensagens na rede e da existência de muitos pares de sincronização. Todos estes aspectos devem ser investigados individualmente.

Neste capítulo foi explicado que para realizar experimentos de medição em rede como a *Internet*, muitas vezes, também se faz necessário a utilização de um relógio externo para a sincronização das máquinas. A precisão dessa referência externa varia conforme a natureza da rede que está sendo avaliada. Tipicamente para redes de alta velocidade deve ser da ordem de milisegundos.

Foram apresentados neste capítulo algoritmos de sincronização, interna e externa, baseados em trocas de mensagens em rede. Dentre estes, os algoritmos de Cristian e Berkeley possuem a grande desvantagem de serem destinados ao uso de uma *Intranet*. Por outro lado o NTP é um serviço de sincronização baseado em uso para *Internet* e que possui grande eficiência, pois apresenta 5 tipos de algoritmos que se destinam não só a acertar a hora local de cada máquina, mas também a disciplinar seus relógios. Este serviço também apresenta melhor robustez e escalabilidade, o que é resultante da redundância dos caminhos de disseminação.

Capítulo 5

Ferramentas para Geração e Medição de Tráfego

5.1. Introdução:

A geração de tráfego sintético possui várias utilidades como, por exemplo, a medição de redes para conhecer as suas características de desempenho e qualidade, e a coleta de dados históricos e em tempo real para atividades de engenharia de tráfego.

Nos últimos tempos, muitas infra-estruturas de medição de redes foram propostas como forma de avaliar eficientemente o desempenho das redes de telecomunicações. Apesar das vantagens apresentadas por essas infra-estruturas, a sua utilização em larga escala ainda é problemática. Isso ocorre principalmente devido a problemas de escalabilidade no gerenciamento de equipamentos de diferentes domínios administrativos.

Cada vez mais as simulações de geração de tráfego devem refletir não apenas a grande escala dos cenários reais, mas também a enorme variedade das fontes de tráfego, em termos tanto de protocolos como de padrões de geração de dados. A simulação de tráfego de camada 4 a 7, sejam eles VoIP, FTP, Telnet, DNS , dentre outros, pode auxiliar na configuração de redes de alto desempenho.

5.2. Ferramentas de Geração e Medição

As ferramentas de geração e medição de tráfego variam desde comandos incluídos nos sistemas operacionais como *ping* [ICMP81] e *Traceroute* [TCR89], passando por ferramentas (*softwares*) gratuitas de código aberto, até sistemas e pacotes comerciais. A grande diferença entre essas, é a limitação das medições a serem utilizadas. Os comandos incluídos nos sistemas operacionais se limitam a medir o retardo unidirecional ou de ida-e-volta através do envio de pacotes ICMP [ICMP81]. Já as ferramentas (*softwares*) possuem uma série de artifícios para gerar tráfego em uma rede, podendo caracterizar determinados tráfegos de acordo com um perfil previsto.

Como parte das atividades desenvolvidas para esta dissertação, foi realizado um estudo comparativo entre as ferramentas disponibilizadas publicamente para a geração de tráfego de rede. Para tal, foram utilizados como referências documentações confiáveis como meio de obter detalhes sobre critérios e características relevantes, para que seja feita uma comparação das ferramentas de forma mais precisa.

5.3. Avaliação do software

A norma NBR 13596 [NBR96] define características que descrevem a qualidade de um *software*. São elas:

- a) Funcionalidade - Conjunto de atributos que evidenciam a existência de funções e suas propriedades especificadas. As funções são as que satisfazem as necessidades explícitas ou implícitas do usuário;
- b) Usabilidade - Conjunto de atributos que evidenciam o esforço necessário para se poder utilizar o *software*, bem como o julgamento individual desse uso, por um conjunto explícito ou implícito de usuários;
- c) Manutenibilidade - Conjunto de atributos que evidenciam o esforço necessário para fazer modificações específicas no software;
- d) Portabilidade - Conjunto de atributos que evidenciam a capacidade do software de ser transferido de um ambiente para outro.

5.4. Critérios de Comparação

O critério de comparação utilizado foi o de buscar nas ferramentas, características para atuar em testes de desempenho de rede de alta velocidade, bem como das aplicações por elas suportadas.

5.5. Ferramentas Analisadas

Para avaliar as ferramentas, o ponto de partida foi a documentação disponível através da página dos desenvolvedores na Internet. Das ferramentas encontradas na *Internet*, foi verificado que muitas tiveram seu uso descontinuado, e por isto não entraram no quadro de comparação. O segundo passo foi instalar cada uma das ferramentas através do sistema operacional *Linux Slackware* 11.0, a fim de verificar características de configuração e o modo como cada ferramenta funciona, além de buscar maiores detalhes sobre cada ferramenta.

5.5.1. TMG

O *Traffic Monitor Generator* - TMG [TMG08] é um software desenvolvido pela Universidade Federal do Rio de Janeiro (UFRJ).

O objetivo do TMG é medir o atraso de ida-e-volta de pacotes e detectar possíveis variações estatísticas deste retardo (*jitter*). Essa ferramenta utiliza o esquema monitor/refletor para cálculo dessas estatísticas, sendo otimizada para medidas de desempenho de alta precisão.

A ferramenta é composta de quatro programas: Gerador, Monitor, Refletor e *SendWait*, que podem ser utilizados em conjunto ou separadamente, e possuem ajuda na linha de comando. Os programas desenvolvidos para as pilhas UDP/IP e TCP/IP fazem uso de programação *socket* BSD padrão, com versões para os sistemas operacionais Solaris, *Linux* e *FreeBSD*.

Cada variante do programa monitor (UDP, TCP, ATM), exceto a variante TCP, apresenta dois tipos de chamadas:

Monitor: mais adequado a sessões de pequena duração, quando se deseja uma medição rápida dos parâmetros de latência do ambiente de rede sendo avaliada.

Monitor2: é otimizado para sessões de maior duração, ideal para situações onde se deseja determinar possíveis variações estatísticas no comportamento da rede, durante um período mais longo de observação.

Os programas refletores têm como única função a leitura e subsequente retransmissão das PDU's enviadas pelos programas monitores equivalentes, sem qualquer processamento extra de informação. Já os programas geradores têm por objetivo a geração de tráfego de fundo, durante a realização dos experimentos. Esses programas permitem a geração de grandes volumes de tráfego (dependendo da capacidade da estação sendo utilizada), sendo capazes de congestionar ambientes de rede local como *Fast Ethernet*, permitindo simulações de uma rede com carga.

5.5.2. MGEN

O *Multi-Generation* - Mgen [MGE08] é um conjunto de ferramentas desenvolvidas pelo Laboratório de Pesquisa Naval [NRL047] dos EUA. Esta ferramenta permite gerar e medir tráfego UDP, tanto unicast quanto multicast. Possui suporte para o protocolo RSVP (*Resource Reservation Protocol*) e permite a criação de scripts utilizando o campo TOS (*Type of Service*) do cabeçalho IP. É multi-plataforma, sendo atualmente suportado pelos seguintes sistemas operacionais: SunOS 4.1.x e 2.x, Linux, Solaris-i386, NetBSD e FreeBSD.

Essa ferramenta é constituída basicamente por duas aplicações: Mgen e Drec e alguns programas utilitários. O Mgen gera padrões de tráfego, e o Drec (*Dynamic-receiver*) recebe e armazena o tráfego gerado pelo programa Mgen. Os utilitários: Mcalc (*Multi-calculator*) analisa os arquivos criados pelo Drec e mostra estatísticas do tráfego recebido; o Rcalc é um visualizador gráfico dos níveis de tráfego por fluxo; o Allcalc gera fluxo estatístico incluindo a taxa média de pacote, a taxa média de bit e a taxa média de atraso. Já o script Ez através da execução do utilitário Allcalc, gera um arquivo em formato gif para ser utilizado pela ferramenta gnuplot e o Txdelay produz um arquivo texto do atraso de transmissão de acordo com o tempo.

A geração do tráfego é realizada através de um arquivo de *script* que emula padrões de tráfego de aplicações UDP/IP unicast ou/e multicast de forma periódica ou através de uma função de POISSON. Isso torna possível a quantificação da taxa de transmissão de pacotes por segundo e do tempo de duração da geração de pacotes. Antes de executar o Mgen, é necessário executar a ferramenta Drec na máquina cliente através da linha de comando, e este é o responsável por criar um relatório com as informações do tráfego recebido.

Existem duas formas de visualização do tráfego: em modo texto ou modo gráfico. Em modo texto, deve-se executar o utilitário Mcalc para a geração de relatório que apresenta estatísticas do tráfego recebido separado por fluxo, e o sumário de todos os fluxos transmitidos. Nessas estatísticas são fornecidas informações sobre a identificação do fluxo, a origem e o destino do fluxo, quantidade de pacotes recebidos, taxa de pacotes e dados recebidos, taxa média, máxima e mínima do atraso, a variação do atraso e o número de pacotes descartados. Para

a visualização gráfica do tráfego, deve-se executar o script Ez, que através da ferramenta gnuplot gera estatísticas por taxa média de pacote, taxa média de bits, e da taxa média de atraso.

5.5.3. Rude/Crude

O Rude/Crude [RUD08] é composto de duas ferramentas que foram desenvolvidas pela Universidade de Tampere, na Finlândia.

O Rude é executado através de um arquivo de configuração em formato texto que permite configurar os fluxos a serem gerados e modificar o comportamento desses fluxos depois de iniciada sua transmissão. Porém, essa ferramenta só é capaz de gerar e medir tráfego UDP.

O Rude permite também o assinalamento do byte TOS do cabeçalho IP, e a criação de novos tipos de fluxo e tráfego. Não existe opção de execução dos parâmetros utilizados no arquivo de script através da linha de comando.

O Crude é a ferramenta que recebe os dados gerados pelo Rude e, na sua forma padrão, exibe informações sobre o tráfego recebido na tela, permitindo o seu redirecionamento para um arquivo em binário, otimizando as operações de I/O. São armazenadas neste arquivo, informações sobre a identificação do fluxo, o número de seqüência do pacote transmitido, o tempo de transmissão e recepção, e o tamanho do pacote em bytes.

Apesar de não fazer parte do pacote do Rude/Crude, foi desenvolvido um utilitário, também pela Universidade de Tampere, chamado qosplot, que permite a conversão dos arquivos gerados pelo Crude. O qosplot é uma ferramenta que utiliza os arquivos gerados pelo Crude como entrada e gera arquivos de dados e de comandos para o utilitário gnuplot. Através deste utilitário, é gerado um arquivo em formato HTML, onde são apresentados diversos gráficos com características de QoS com as informações de percentual de perda de pacotes, vazão, latência, variação da latência, além de um sumário destas características.

5.5.4. IPERF

O IPERF [IPF08] é um projeto desenvolvido pelo *National Laboratory for Applied Network Research - NLANR* [NLAN08], fundado pela *National Science Foundation - NSF*. É um *software* com características para análise de desempenho de redes, que gera tráfegos com perfis variados tanto para o TCP quanto para o UDP.

Este software foi desenvolvido para ser uma alternativa moderna para a medição de desempenho e utilização de banda de conexões TCP e UDP. Como a maioria das ferramentas de

geração de tráfego, o IPERF utiliza o modelo cliente/servidor, onde o servidor é capaz de manipular múltiplas conexões ao invés de encerrar a execução após um único teste.

Para realizar as medições, o IPERF envia pacotes do cliente de forma bastante rápida. A informação é enviada diretamente da memória do cliente para a memória do servidor. Porém, para redes de alta velocidade, frequentemente é necessário utilizar múltiplos fluxos para maximizar a banda.

O tempo de geração do tráfego pode ser determinado através de um intervalo específico, ou da quantidade de dados a ser transmitida. Ao final do experimento, ou periodicamente em intervalos especificados, são gerados relatórios de perda de pacotes, jitter, e banda. São relatórios em texto, utilizando a unidade de dados mais adequada.

5.5.5. DITG

O *Distributed Internet Traffic Generator* – DITG [DITG08] é uma ferramenta criada pelo Departamento de Informática da Universidade de Nápoli na Itália, cuja proposta é poder ser utilizada facilmente através de um conjunto de experimentos que possam ser repetidos, utilizando uma mistura confiável e realista dos tipos de tráfego disponíveis.

O DITG pode gerar tráfego de acordo com várias distribuições de probabilidade. Esta ferramenta foi desenhada para a geração de tráfego de camada de aplicação, suportando os seguintes protocolos: TCP, UDP, ICMP, DNS, Telnet, VoIP (G.711, G.723, G.729, *Voice Activity Detection*). Com o DITG é possível a reprodução de condições de rede muito complexas, sob diferentes cargas e configurações de tráfego, permitindo também a investigação de efeitos de escala. Esta ferramenta é capaz de gerar tráfego a altas taxas, sendo bastante adequada para redes funcionando em alta velocidade.

O DITG permite que tanto a duração do experimento, como o retardo em relação ao tempo de início do experimento, possam ser configurados.

Para analisar os resultados dos experimentos é utilizado o ITGDec, um decodificador desenvolvido para a plataforma de geração DITG. A partir dos arquivos de log gerados pela origem (ITGSend) e pelo destino (ITGRecv), o ITGDec calcula os valores médios da taxa de transmissão, retardo e *jitter* de todo o experimento, ou de um intervalo específico.

Embora não faça parte do pacote de instalação, o DITG permite utilizar o software Octave [OCT08]. Este é o responsável por transformar o arquivo decodificado pelo utilitário Drec, em um arquivo gráfico, disponibilizando informações sobre velocidade de transmissão, atraso, variação de atraso e quantidade de pacotes perdidos durante um intervalo de tempo especificado.

5.6. Comparação das ferramentas

A seguir estão tabuladas as informações mais importantes das ferramentas avaliadas para que seja possível sua comparação de forma simples e sistemática. Não é feito nenhum julgamento quanto a melhor ferramenta, uma vez que cada uma pode ser a mais indicada para determinado tipo de medição e de ambiente, porém como o foco deste capítulo está voltado para situações de testes para análise de desempenho de uma rede de alta velocidade, o objetivo é o de buscar ferramentas que melhor satisfaçam estas características.

Os itens (F), (M), (P) e o (U) representam respectivamente, funcionalidade, manutenibilidade, portabilidade e usabilidade. Estes itens estão presentes na primeira coluna das tabelas seguintes, representando as características de qualidade de software definidas pela NBR 13596 [NBR96].

Os critérios de julgamento das comparações mostradas nas tabelas foram definidos em função dos testes da caixa-preta que tem como objetivo testar as interfaces de entrada e saída. Quando um item é marcado com o caracter “√”, significa que o mesmo encontra-se presente na ferramenta avaliada, caso contrário, o item será marcado com o caracter “×”.

As informações, que são na sua maioria auto-explicativas ou se referem a métricas já discutidas no capítulo 3, estão divididas em blocos de acordo com a sua utilização e/ou seu tipo.

5.6.1. Características gerais das ferramentas

Quadro 2: Características de funcionamento

Produtos	NBR	DITG	IPERF	MGEN	Rude	TMG
Modo de Entrada						
Arquivo de configuração (Script)	U	✓	×	✓	✓	×
Comand Line Interface (CLI)	U	✓	✓	✓	×	✓
Interface Gráfica	U	×	×	✓	×	×
Característica						
Ipv6	F	×	✓	✓	×	×
RSVP	F	×	×	✓	×	×
TOS	F	✓	×	✓	✓	×
Escuta de múltiplas portas	F	✓	✓	✓	×	×
ATM nativo	F	×	✓	×	×	✓
Fontes	M	✓	✓	✓	✓	×
Sistema Operacional						
Sun	P	×	×	✓	×	×
Solaris	P	×	×	✓	✓	✓
Linux	P	✓	✓	✓	✓	✓
NetBSD	P	×	✓	✓	×	×
FreeBSD	P	×	×	✓	×	✓
Win32	P	✓	✓	✓	×	×
HP	P	×	✓	×	×	×
Irix	P	×	×	×	×	×
Digital Unix	P	×	×	×	×	×

De acordo com o Quadro 1, pode-se observar que a Mgen é a ferramenta que possui o maior número de características necessárias para o funcionamento em um ambiente onde aplicações possam requerer certa qualidade de serviço.

Quadro 3: Tipos de tráfego

Produtos	NBR	DITG	IPERF	MGEN	Rude	TMG
TCP	F	✓	✓	×	×	✓
UDP	F	✓	✓	✓	✓	✓
ICMP	F	✓	×	×	×	×
VoIP	F	✓	×	×	×	×
Telnet	F	✓	×	×	×	×
DNS	F	✓	×	×	×	×
Multicast	F	✓	✓	✓	×	×
Em Rajadas	F	✓	✓	✓	✓	✓
Tráfego de Fundo	F	×	×	×	×	✓

No Quadro 2 as ferramentas que apresentaram maior flexibilidade quanto aos tipos de tráfego existentes foram a DITG e a IPERF.

Quadro 4: Parâmetros de entrada (UDP)

Parâmetros	NBR	DITG	IPER F	Mgen	Rude	TMG
Hora de início da transmissão do fluxo	F	×	×	✓	×	×
Help	U	✓	✓	✓	✓	✓
Endereço destino	F	✓	✓	✓	×	✓
Porta destino	F	✓	✓	✓	×	✓
Tamanho do pacote(bytes)	F	✓	✓	✓	✓	✓
Taxa de pacotes	F	✓	✓	✓	✓	✓
Intervalo de transmissão entre pacotes	F	✓	×	×	×	✓
Intervalo de transmissão entre rajada	F	✓	✓	✓	✓	✓
Duração da sessão	F	✓	✓	✓	✓	✓
Especificação do Meio físico	F	×	×	×	×	✓
Suporte a Múltiplas CPU	F	×	✓	×	×	×
Total de pacotes a serem gerados	F	✓	×	✓	✓	✓

O Quadro 3 mostra os parâmetros de entrada para tráfego UDP de cada ferramenta analisada, indicando que há características de funcionalidades que estão disponíveis em uma única ferramenta. Por exemplo, o parâmetro “Especificação do meio físico”, que somente está presente na ferramenta TMG, e é utilizado para cálculo das taxas efetivas geradas para o meio físico no final da transmissão, ou o parâmetro de “Hora de Início de Transmissão” utilizada pela ferramenta Mgen, que determina a hora exata de início da transmissão do fluxo de teste.

Quadro 5: Parâmetros de entrada (TCP)

Parâmetros	DITG	IPERF	TMG
Endereço destino	✓	✓	✓
Porta destino	✓	✓	✓
Tamanho do pacote (Bytes)	✓	✓	✓
Intervalo entre transmissões consecutivas (us)	×	×	✓
Duração da sessão em segundos	×	✓	×
Duração da sessão em bytes	✓	✓	×
Suporte a Múltiplos Processadores	×	✓	×
Número de pacotes a serem gerados	✓	✓	✓
Intervalo entre rajada	✓	✓	×
Taxa de pacotes transmitidos por segundo	✓	×	×

O Quadro 4 mostra os parâmetros de entrada para tráfego TCP de cada ferramenta analisada. Como o Rude/Crude e Mgen não possuem parâmetros para emular tráfego TCP, não foi utilizada esta tabela na composição das características dada às ferramentas analisadas, visto que no Quadro 2, foi considerado como uma característica geral o fato da ferramenta gerar tráfego UDP.

A ferramenta IPERF é a mais rica nos parâmetros de entrada, para a realização de experimentos utilizando um perfil de tráfego específico para o TCP.

Quadro 6: Informações de saída

Informações	NBR	DITG	IPERF	Mgen	Rude	TMG
			F			
Identificação do fluxo	U	✓	×	✓	✓	✓
Identificação dos pares de máquina	U	✓	×	✓	×	×
Separação por fluxo	U	✓	×	✓	✓	✓
Qtd. de pacotes (blocos)RX	F	✓	✓	✓	✓	✓
Qtd. de pacotes (blocos)TX	F	✓	✓	✓	✓	✓
Média de atraso	F	✓	×	✓	✓	✓
Atraso	F	✓	×	✓	✓	✓
Tempo de resposta	F	×	✓	×	×	×
Vazão	F	✓	✓	×	×	×
Descarte total	F	✓	×	✓	✓	✓
Taxa de transmissão	F	✓	✓	✓	✓	✓
Variação de atraso (Jitter)	F	✓	✓	✓	✓	×

O Quadro 5 mostra informações dos parâmetros de saída dos relatórios de cada ferramenta analisada. Através dessas, pode-se observar que somente a ferramenta TMG não

fornece informações de variação de atraso no relatório gerado, sendo essa funcionalidade considerada importante para uma ferramenta com características de testes de desempenho.

5.7. Justificativa de escolha das ferramentas

Todas as ferramentas em estudo ofereceram facilidade de instalação e uso, não necessitando de configurações adicionais ao sistema operacional.

As ferramentas foram analisadas segundo a norma NBR 13596 [NBR96] e procurou-se selecionar ferramentas que tivessem perfil para testes de desempenho, e que por sua vez pudessem funcionar com várias aplicações distribuídas, já que a simulação de diferentes tráfegos como VoIP, Telnet, ICMP, dentre outros, pode auxiliar na geração de tráfego para redes de alto desempenho, permitindo a coleta contínua, e em tempo real, da qualidade e de condição da mesma.

Baseando-se nos critérios de comparação descritos anteriormente para seleção das ferramentas, a primeira ferramenta escolhida foi o DITG, por esta permitir a possibilidade de gerar diferentes perfis de tráfego, controlando seus parâmetros e simulando as principais aplicações utilizadas em uma rede IP. Também foi considerada a taxa máxima de geração desta, visando especificamente a geração de tráfego a taxa de Gbps.

A segunda ferramenta escolhida foi a IPERF, pela capacidade que esta tem de obter diversas medidas ativas, de forma a permitir a avaliação de desempenho de redes, incluindo enlaces, equipamentos e serviços.

Veja a seguir as principais características observadas nas ferramentas escolhidas:

- a) IPERF: Esta ferramenta apresentou boas características de funcionalidade e usabilidade, sendo toda criada com base em testes de medição de desempenho de enlace de rede, além de possuir a melhor riqueza de parâmetros de entrada para emular tráfego TCP. É uma ferramenta que foca basicamente dois tipos de testes: desempenho e tempo de resposta e, conseqüentemente, os resultados apresentam a quantidade de pacotes transmitidos e recebidos, a taxa de transmissão e informações sobre desempenho (*throughput* e *response time*). É uma ferramenta muito utilizada pela comunidade de pesquisa internacional, sendo, portanto, bem aceita em termos de confiabilidade de resultados.
- b) DITG: Esta ferramenta tem como característica principal gerar um conjunto de experimentos que possam ser repetidos utilizando uma mistura bastante realista dos tipos de tráfego disponíveis para uma aplicação. Com o DITG é possível também a

investigação de efeitos de escala para os diferentes tráfegos gerados, sendo bastante útil para experimentos que envolvam engenharia de tráfego.

5.8. Projetos de Medição Ativa

Existem diversos projetos que fazem medições ativas de desempenho fim a fim na Internet (AIEPM – *Active Internet End-to-end Performance Measurements*). A principal diferença desses projetos para as ferramentas avaliadas no item anterior é sua utilização específica na *Internet* e a grande abrangência geográfica, distribuindo as tarefas de medições entre um grande número de pontos. Outra diferença importante é a limitação em definir os parâmetros de transmissão e dos pacotes de teste. Na maioria desses projetos, a medição é baseada em ferramentas já conhecidas, como *ping* [ICMP81] e *traceroute* [TCR89], ou na implementação das recomendações do IETF [IETF08].

5.8.1. CAIDA

A *Cooperative Association for Internet Data Analysis* - CAIDA [CAID08] é uma associação colaborativa com participantes dos setores comercial, governamental e acadêmico. O principal objetivo dessa associação é a investigação da estrutura e do comportamento da *Internet* através de medições para uma melhor compreensão da rede para extensão a níveis globais. Dessa forma, os objetivos se dividem em medições e análise da infra-estrutura da Internet, investigação de novas tecnologias para melhorar o desempenho da rede, caracterização do comportamento do tráfego presente na rede através de medições passivas e ativas, e o desenvolvimento de ferramentas para análise e visualização de características da rede.

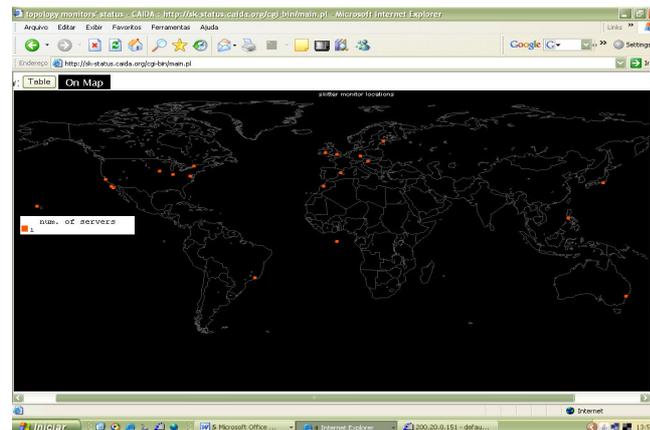
Atualmente a CAIDA vem se concentrando no desenvolvimento de ferramentas para a medição, análise e visualização de dados da Internet. Alguns exemplos de ferramentas são a CoralReef, criada para a análise de dados gerados por pontos de medição passiva, e a Skitter criada para a coleta de dados de forma a obter o mapeamento de estrutura da Internet. Atualmente mais de 23.000 destinações são monitoradas a partir de 20 pontos distribuídos nos EUA, Europa e Ásia.

5.8.1.1. Skitter

Skitter [SKT08] é uma ferramenta que propõe analisar topologia e desempenho na Internet. Esta ferramenta determina o caminho unidirecional entre a sua localização, definida como origem, e um ou mais destinos, utilizando um método semelhante ao *traceroute*, ou seja,

testando cada enlace ao longo do caminho através do envio de vários pacotes ICMP e incrementando o campo TTL (time-to-live) do cabeçalho IP. Dessa forma, são identificadas mudanças de roteamento persistentes. As medições ativas realizadas são: retardo de ida e volta, topologia e variação do atraso (Jitter). A figura 5 desmonstra os pontos de atuação dos servidores Skitter.

Figura 6: Mapa dos servidores Skitter. [SKT08]



5.8.2. AMP – *Active Measurement Program*

O AMP é um projeto desenvolvido pelo laboratório Nacional de Redes Aplicadas [NLAN08]. Atualmente a malha de medição do AMP conta com 150 monitores instalados pelo mundo.

Os monitores AMP são compostos, basicamente, por medidores denominados de *Amplets* e coletores ou centralizadores de dados chamados de *DataCollectors*. Os *amplets* utilizam o *fping* para enviar pacotes ICMP para múltiplos *hosts* simultaneamente, enquanto os *DataCollectors* coletam os dados que foram coletados pelos *Amplets* e os armazenam em seus discos emitindo os resultados na forma de relatórios.

Amplet é um medidor que faz parte da estrutura de um monitor AMP. Um *Amplet* faz basicamente a medição de atraso de ida e volta e perda de pacotes a cada minuto e com uma variação aleatória de quinze segundos de espera, e da medição da topologia a cada dez minutos e com variação aleatória de quinze segundos de espera. Utilizando a ferramenta *traceroute* um *Amplet* conhecido dos outros *amplets* se comunica fazendo as medições direcionadas a um ao outro, através de uma configuração interna de uma lista de *Amplets* conhecidos.

O *DataCollector* ou simplesmente “Coletor de Dados” somente recebe dados de medições gerados pelos *Amplets*. Feita a coleta dos dados, o *DataCollector* os armazena e gera os gráficos e estatísticas a partir de solicitações feitas pelos usuários para visualização dos relatórios através da interface *web* [WIKI08]. No momento em que esses relatórios são

requeridos pelo usuário, estas requisições geram um *overhead* grande no sistema e podem interferir nas medições realizadas. Dessa forma, a divisão entre *Amplets* e *DataCollector* se faz necessária tanto para não haver interferência nas medições como também para a centralização dos dados num ponto distinto.

5.8.3. Surveyor

O *Surveyor* [SUR08] é um projeto criado por duas organizações ligadas à pesquisa: *Advanced Network Services* – ANS [ANS08] e *Common Solutions Group* – CSG [CSG08]. Seu principal objetivo é criar tecnologia e infra-estrutura de medição permitindo aos usuários e provedores de serviços terem um entendimento preciso do desempenho e da confiabilidade para determinados caminhos na Internet. Permite também distinguir qual segmento causa limitações, utilizando testes ativos unidirecionais ao longo de caminhos entre máquinas dedicadas à realização de medições. Essas máquinas ficam localizadas em alguns *sites* associados. A comunidade de interesse são os centros de pesquisa e de educação superior dos Estados Unidos, ligados ao CSG e à ANS.

O *Surveyor* utiliza as métricas padronizadas pelo grupo de trabalho IPPM do IETF [IPPM08]. As medições ativas realizadas são: retardo e perda de pacotes.

5.9. Comparação dos Projetos Avaliados

Os projetos podem ser divididos entre o tipo de medição que executam (unidirecional ou ida e volta), a necessidade de uma máquina dedicada para as medições, o tipo de sincronismo que utilizam entre as máquinas, o tamanho dos pacotes usados para as medições, e os locais de atuação dos projetos. Essas informações estão tabuladas na Tabela 1 a seguir, para que seja possível a comparação entre os projetos apresentados.

Tabela 1: Comparação dos Projetos de Medição Ativa

	Surveyor	AMP	Skitter
Método	Recomendações do IETF	Ping	Traceroute
Máquina	Dedicada	Dedicada	Dedicada
Sincronização	GPS	NTP	NTP
Tamanho do pacote	40 Bytes	64 Bytes	53 Bytes
Locais de Atuação	US, CA, CH, NL e NZ	US, NZ, NO	Ásia, CA, UK e US

A base do projeto *Skitter* possui um diferencial em relação aos demais, que é o de buscar mapear a Internet como um todo através de um grande volume de medições, sendo considerado um projeto mais genérico, não direcionando seu monitoramento a nenhum fim específico. Já o *Surveyor* é o projeto que realiza medidas mais frequentes, porém não disponibiliza seus dados para o acesso do público.

5.10. Conclusão

Através do estudo comparativo de ferramentas de geração de tráfego realizado neste capítulo, observa-se que dependendo das características implementadas em um ambiente de teste, determinadas ferramentas são mais atrativas que outras. Porém, como todo o foco deste trabalho está voltado para a reprodução de condições de redes complexas sob diferentes cargas de tráfegos, foram escolhidas apenas as ferramentas IPERF e DITG.

A maioria dos projetos de medição ativa descritos através deste capítulo trabalha com medições voltadas para uso de *ping* [ICMP81] ou *traceroute* [TCR89], o que torna alguns resultados duvidosos, pois medições como perda de pacotes no projeto AMP, é definida pelo não recebimento de quatro respostas ao ICMP, ficando difícil caracterizar o congestionamento, já que esta perda geralmente ocorrerá quando houver falhas na rede, não sendo uma ferramenta atualmente muito adequada para a detecção de congestionamento. Além do mais, estes projetos inviabilizam medições destinadas a estudos específicos, já que não permitem a definição do tipo de tráfego a ser configurado para uma determinada medição.

Capítulo 6

SCGT: Sistema de Geração de tráfego Distribuído

6.1. Introdução

Os ambientes que usam diversas ferramentas de geração de tráfego sintético têm como principal dificuldade a carência por infra-estruturas de coleta que sejam flexíveis e adaptáveis às aplicações dos usuários finais, bem como às ferramentas que fazem uso dos dados de medições. Essas razões levam a uma forte tendência para a utilização de Serviços *Web* nas infra-estruturas de medições, como forma de reduzir os problemas de interoperabilidade das aplicações. Isso faz com que cada ferramenta exponha as suas funcionalidades de forma bem definida e em forma de serviços, utilizando a *Web* [WIKI08] como o principal meio de comunicação.

Apesar de algumas ferramentas destinadas ao uso de medição ativa serem bastante utilizadas, podendo ser encontradas livremente pela *Web*, poucos trabalhos significativos foram realizados em torno da visualização dos tráfegos gerados, e do sincronismo realizados por estes.

Com base nessas informações, foi projetado o SCGT, um Sistema de Coordenação para Geradores de Tráfego Sintético. Este sistema permite gerar tráfegos dos mais variados perfis, realizando testes simultâneos e agendados entre as máquinas envolvidas em experimentos de medição ativa. O SCGT tem como principal objetivo medir características de desempenho na rede GIGA, realizando medições na *Web* [WIKI08] através do uso de uma interface gráfica. Entre as coletas, podem-se observar informações sobre as condições da rede e de seus vários elementos. Dentre as métricas utilizadas nas coletas estão: latência, *jitter*, perda de pacotes e vazão.

6.2. Descrição Geral

O Sistema de Coordenação para Geradores de Tráfego Sintético (SCGT) foi desenvolvido para integrar as ferramentas de geração de tráfego IPERF e DITG, coordenando seus experimentos, de acordo com o modo como estes são realizados. Este sistema possui uma interface gráfica totalmente desenvolvida em php [PHP08] e *javascript* [JAVA08], utilizando técnicas de DHTML [DHTM08]. Dessa forma, as ferramentas de geração de tráfego podem ser disparadas em conjunto, através de qualquer máquina que esteja integrada a rede GIGA.

Este sistema foi criado a partir da limitação de algumas funcionalidades encontradas nas ferramentas de geração de tráfego já existentes, e que são consideradas de bastante relevância para um ambiente distribuído de medição ativa.

6.3. Serviços

O SCGT oferece serviços de experimentos utilizando geradores de tráfego, adicionando a estes o modo como é tratado cada experimento e a forma como são gerados os tráfegos. Sendo assim, o experimento pode ser inicializado do seguinte modo:

- ✓ Modo instantâneo – Neste serviço os experimentos são inicializados dentro de um tempo determinado pelo sistema.
- ✓ Modo agendado - Este serviço é uma inovação do SCGT criado para atender aos usuários deste sistema, já que é considerada uma característica relativa à usabilidade do software segundo a norma NBR [NBR96]. Através deste, o usuário tem o direito de escolher a hora que deseja realizar os experimentos, e este pode ser inicializado no momento previsto pelo usuário.

O usuário pode combinar estes modos de inicialização de cada experimento, com a forma como deseja iniciar (gerar) o tráfego. Dentre elas:

- ✓ Forma Não Simultânea – Neste serviço, um único fluxo é inicializado, através de um único experimento.
- ✓ Forma Simultânea – Este serviço também é uma inovação do SCGT, considerado como uma característica relativa à usabilidade do software segundo a norma NBR [NBR96]. Neste caso, diferentes fluxos podem ser inicializados ao mesmo tempo e através de um único experimento.

Outra inovação do SCGT é o serviço de armazenamento dos relatórios, onde o relatório de cada teste anteriormente executado, é armazenado em um repositório de dados e o usuário poderá visualizá-lo futuramente.

6.4. Produtos

A cada novo experimento feito, o SCGT é capaz de apresentar, através da interface gráfica, relatórios textos contendo os detalhes do tráfego medido pelas ferramentas de geração de tráfego, além de produzir e adicionar a apresentação de:

- **Relatórios Gráficos:** Característica relativa à usabilidade do *software* [NBR96]. O SCGT acrescenta a apresentação de relatórios gráficos medidos através da ferramenta DITG. São relatórios obtidos através do *software Octave* [OCT08], que permite decodificar o relatório produzido pelo DITG para o formato gráfico, apresentando estimativas sobre atraso, variação do atraso, velocidade e quantidades de pacotes perdidos ao longo de um experimento.

O tempo de recebimento do relatório é indiretamente estabelecido pelo próprio usuário, quando este determina o modo como vai ser realizado o experimento. Os relatórios gerados através de experimentos realizados no modo instantâneo são obtidos automaticamente após o término do experimento. Já para experimentos realizados no modo agendado, os relatórios são obtidos no tempo computado pelo sistema.

6.5. Composição do Sistema

O SCGT é composto por um módulo de coordenação para Geradores de Tráfego (CGT) responsável por interagir com a interface gráfica, e gerar os produtos através da manipulação das ferramentas IPERF e DITG. Para tal este dispõe de um conjunto de funções, reunidas, que:

- ✓ configure experimentos a serem feitos;
- ✓ manipule as ferramentas;
- ✓ gere informações adicionais;
- ✓ formate o relatório final para divulgação.

Os usuários do SCGT devem informar através da interface gráfica, o tipo de experimento a ser realizado. Cada novo serviço é processado pelo módulo CGT e coordenado de acordo com o que fora estabelecido pelo usuário, e cada novo produto é automaticamente incluído no repositório de dados de dados do sistema, e divulgado para todos os usuários que tiverem demonstrado interesse na visualização do mesmo.

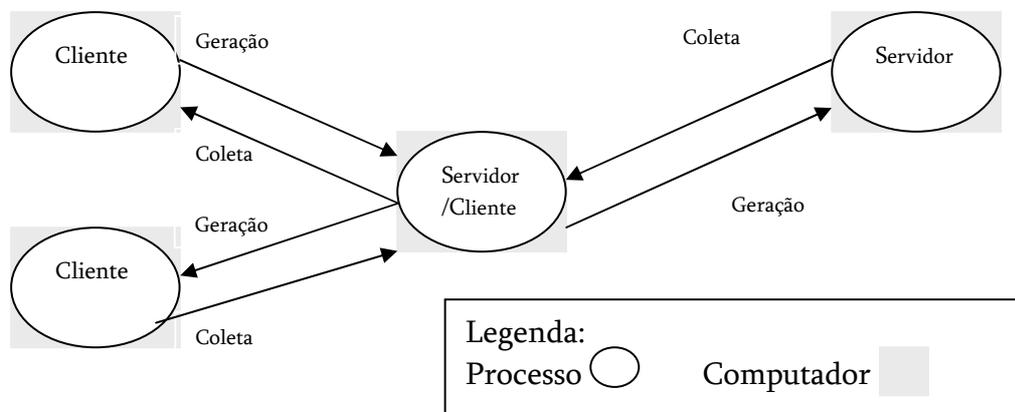
6.6. Aspectos relacionados ao funcionamento

A arquitetura cliente-servidor é bastante utilizada em redes [COUL08], onde os servidores podem por sua vez ser cliente de outros. Por exemplo, o servidor *Web* onde se localiza o sistema SCGT, é cliente de um servidor NTP que administra o sincronismo realizado entre as máquinas.

6.6.1. Modelo de Arquitetura

A interação do módulo CGT com as ferramentas de geração de tráfego é feita através de linhas de *script* nativas das próprias ferramentas. Deste modo, o módulo CGT coordena como vão ser realizados os processos para as máquinas responsáveis por fazer a geração (envio do tráfego) e coleta (medição) do tráfego, e diferentes processos podem funcionar paralelamente em qualquer máquina. Na Figura 7 é ilustrada uma estrutura simples do funcionamento do SCGT na rede GIGA utilizando a arquitetura cliente-servidor, na qual alguns processos entre as máquinas envolvidas nos experimentos podem interagir entre si.

Figura 7: Integração dos Processos



6.6.2. Divisão Estrutural

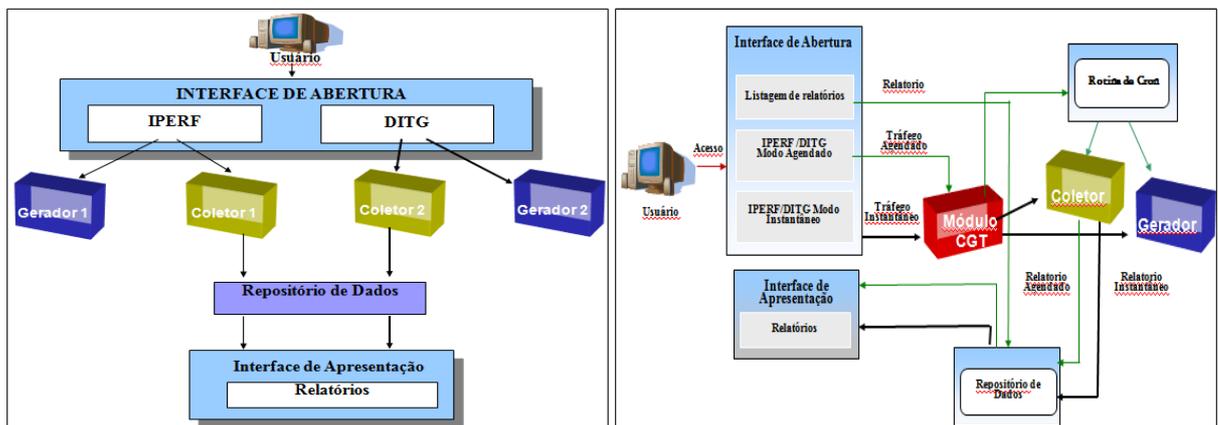
O SCGT atua em um servidor *Web* fazendo a comunicação entre o usuário da rede GIGA e as ferramentas IPERF e/ou DITG. O usuário se comunica com o sistema através de uma interface gráfica, escolhe a ferramenta e o modo como vai ser realizado o experimento (instantâneo ou agendado). Em seguida, este deve preencher ou selecionar as características do tráfego a ser gerado bem como a forma como deseja iniciar o experimento (simultâneo ou não). Ao finalizar a configuração, o módulo CGT imediatamente configura o experimento utilizando os valores atribuídos pelo usuário nas características do tráfego, transformando-o em linhas de *script* para a ferramenta.

De acordo com o modo através do qual vai ser realizado o experimento, o módulo CGT coordena os processos de inicialização dos eventos e abre uma conexão utilizando a arquitetura cliente-servidor entre as máquinas envolvidas no experimento.

Para experimentos a serem realizados no modo agendado, o usuário recebe o relatório final, de acordo com o tempo programado para o envio do tráfego somado ao tempo de duração do experimento. A cópia do relatório estará disponibilizada no repositório de dados. Para experimentos realizados no modo agendado, o usuário recebe o relatório final somente no repositório de dados, de acordo com a hora prevista pelo sistema para o encerramento do experimento. Quando o usuário solicita qualquer relatório final através do repositório de dados, o módulo CGT o interpreta através de atributos de identificação e apresenta-o em tela.

A Figura 8 ilustra este processo, exibindo o modo de comunicação entre o usuário e o SCGT.

Figura 8: Modo de Comunicação



6.6.3. Perfil do Tráfego

O SCGT disponibiliza a configuração dos parâmetros relacionados ao perfil de tráfego, de acordo com a exigência de cada ferramenta. Para a ferramenta IPERF, os perfis são divididos em UDP e TCP. Para a ferramenta DITG, os perfis são divididos em: Personalizado e Pré-Definido.

Um perfil de tráfego pode ser entendido como um conjunto de parâmetros de entrada a serem configurados de acordo com as características apresentadas nativamente por cada ferramenta.

A ferramenta IPERF se destaca por realizar medições relativas ao desempenho da rede, utilizando separadamente diferentes tipos de configuração, tanto para experimentos realizados com o UDP quanto para o TCP. Já a ferramenta DITG, é uma excelente alternativa quando se deseja realizar gerações de tráfegos personalizados, pois permite especificar diferentes parâmetros de entrada a serem configurados em um único experimento, ou então um tráfego pré-

definido, onde alguns parâmetros de configuração são definidos anteriormente pela própria ferramenta.

Sendo assim, cada ferramenta de geração de tráfego sintético contribui ao ambiente de geração e coleta do tráfego de acordo com as suas características, e o SCGT inova e adiciona a estas ferramentas a forma de envio do tráfego e o modo como os experimentos são tratados.

6.6.4. Geração do Tráfego Programada

Para garantir a credibilidade de tráfegos simultâneos, a geração do tráfego é inicializada na hora programada pelo módulo CGT. Dessa forma, o módulo CGT cria um pequeno programa que realiza periodicamente a comparação da hora local da máquina geradora, de acordo com a hora programada para a geração do tráfego.

Este artefato de programação desenvolvido pelo módulo CGT, após ser criado, é executado separadamente através de cada máquina responsável pela geração do tráfego (máquina geradora). Uma vez que este é executado, funciona no máximo durante 60 segundos e o tráfego é enviado com precisão de milisegundos.

Especificamente este pequeno programa para geração do tráfego utiliza a linguagem *Bash* (*Born Again Shell*) do interpretador de comandos *Shell*. A escolha dessa linguagem para sua composição, justifica-se pelo sistema *Linux* possuir por padrão o “interpretador” *shell* nativamente, não precisando de nenhuma instalação e permissão adicional para este poder ser utilizado através das máquinas geradoras. A Figura 9 descreve o programa de geração do tráfego utilizando a ferramenta IPERF.

Figura 9 – Geração de Tráfego Programada

```
#!/bin/bash
#Programa IPERF
#Início do Processo Cliente
#Iniciar geração do tráfego as 13:02:00 horas
contador=1
#Inicia contagem durante 60 segundos
while [ $contador -le 60 ] do
    #Recebe hora atual da máquina geradora
    HATUAL=`date +%H%M%S`
#Se a hora atual da máquina geradora for = 13:02:00
    if [ $HATUAL = "130200" ]; then
#Exemplo do script de geração do tráfego IPERF para a duração de um fluxo de 2 minutos
        IPERF -c10.24.65.9 -w12 -M60 -i1 -t120 >/home/gerador/base/udp_IPERF_2008-05-19_11-12.txt
        fi
# Se a hora atual da máquina geradora não for = 13:02:00
        sleep 1
# Conta 1 segundo para o loop
        contador=`expr $contador + 1`
    done
```

6.6.5. Rotina de Cron

Somente para experimentos a serem inicializados no modo agendado, o módulo CGT configura e utiliza a rotina de cron da máquina onde reside o próprio sistema, agendando o horário de vários processos. A rotina de cron é um serviço presente nativamente no sistema operacional *Linux* e atua como um agendador de processos do SCGT.

5.6.6. Autenticação com *OpenSSH*

Devido à reconhecida segurança na autenticação e confidencialidade dos dados trafegados, o tratamento dos processos a serem executados pelo módulo CGT são disparados através de uma conexão remota criptografada utilizando o aplicativo *OpenSSH* [SSH08]. Este aplicativo é encontrado nativamente no sistema operacional *Unix*. Deste modo, o SCGT irá logar automaticamente em cada máquina, inicializado cada processo.

6.6.7. Ordenação dos Processos

Como definido no capítulo quatro desta dissertação, um sistema distribuído consiste em um grupo de máquinas independentes agrupadas. Dessa maneira, cada qual possui seu relógio interno. Como relógios são processos físicos, eles naturalmente diferem uns dos outros. Entretanto, através de relógios lógicos é possível definir uma ordem total e parcial de eventos.

De acordo com o modo como é inicializado o experimento de geração do tráfego (instantâneo ou agendado), o módulo CGT determina a ordem como são inicializados os processos para que estes possam acontecer de forma sincronizada. No entanto, para não ficar uma leitura exaustiva, só são demonstrados os processos em comum (envio e coleta do tráfego) considerados de suma importância para o sincronismo dos eventos, e outros processos específicos (cópia, decodificação e etc) se encontram no Apêndice 3 desta dissertação.

- ✓ Processo Servidor: É inicializado o “modo servidor” (recebimento do tráfego) da ferramenta, através de conexão remota para a máquina medidora.
- ✓ Processo Cliente: É inicializado o modo “cliente” (geração do tráfego) da ferramenta, através da máquina geradora.

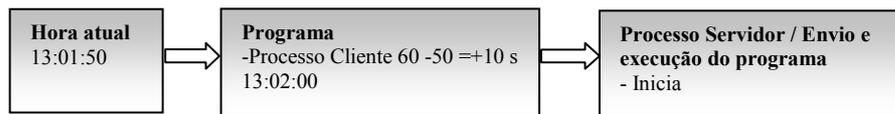
Tanto para o modo agendado quanto para o modo instantâneo, a ordem dos processos é feita da seguinte forma:

1. Programa a hora para o início do processo cliente.
2. Inicia o processo Servidor.
3. Envia o programa de geração do tráfego para a máquina geradora, iniciando-o em seguida.

6.6.7.1. Modo Instantâneo

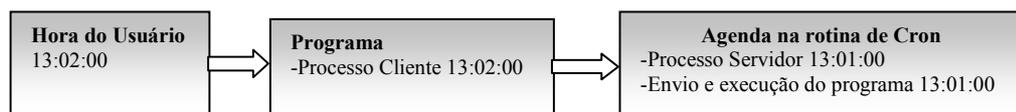
Para tráfegos realizados no modo instantâneo, o módulo CGT registra o processo cliente no programa de geração do tráfego, tomando como base o horário atual, porém adicionando os segundos que faltam para o minuto seguinte e inicia logo após o processo servidor. Ou seja, se o usuário marcou a opção de gerar o tráfego e a hora atual do sistema era 13:01:50, o processo cliente é marcado para as 13:02:00 horas no programa de geração do tráfego, e o processo servidor é imediatamente iniciado.

O programa de geração do tráfego é enviado e executado para a máquina geradora logo após a execução do processo Servidor. A figura 10 representa um exemplo de funcionamento do SCGT no modo instantâneo.

Figura 10: Processos Cliente e Servidor – instantâneo

6.6.7.2. Modo Agendado

Para este modo, o módulo CGT registra a hora do processo Cliente no programa de geração do tráfego, tomando como base o horário proposto pelo usuário e os demais processos são executados pela rotina de cron do sistema com um (1) minuto de antecedência. Ou seja, se o usuário deseja gerar o tráfego as 13:02:00 horas, o módulo CGT configura, no programa de geração do tráfego, o processo Cliente para este horário e, através da rotina de cron, agenda o início do processo Servidor bem como o envio e execução do programa de geração do tráfego para a máquina geradora, no mesmo horário, ou seja, as 13:01:00 horas. Veja o exemplo seguinte ilustrado na Figura 11.

Figura 11: Processos Cliente e Servidor – agendado

Seguindo o exemplo anterior, na Figura 12 é apresentada a configuração da rotina de cron, onde para cada linha, o primeiro par de números representa o minuto e o segundo par representa a hora de início dos processos.

Figura 12 – Configuração da Rotina de cron

```

#Inicializa processo de coleta do tráfego (modo servidor) na máquina medidora
01 13 * * * ssh gerador@10.24.65.9 screen -d -m /usr/src/IPERF-2.0.2/src/IPERF -s
# Inicializa cópia do programa de geração do tráfego e logo em seguida o executa na máquina geradora.
01 13 * * * scp gerador@10.24.65.7 :/home/gerador/base/IPERF /home/gerador/base/; ssh gerador@10.24.65.7
/home/gerador/base/./IPERF
  
```

6.7. Sincronismo dos Experimentos

Quando são feitas medições ativas em uma rede amplamente distribuída, existe um fator considerado de suma importância: o de manter as máquinas envolvidas nos eventos de geração sincronizadas de tal maneira, que uma máquina tenha exatamente a noção de tempo da outra.

Para saber em que hora do dia os eventos ocorrem nos processos em uma rede, é necessário sincronizar os relógios dos computadores com uma fonte de tempo externa de referência. Esta é a sincronização externa e se os relógios são sincronizados com um grau de

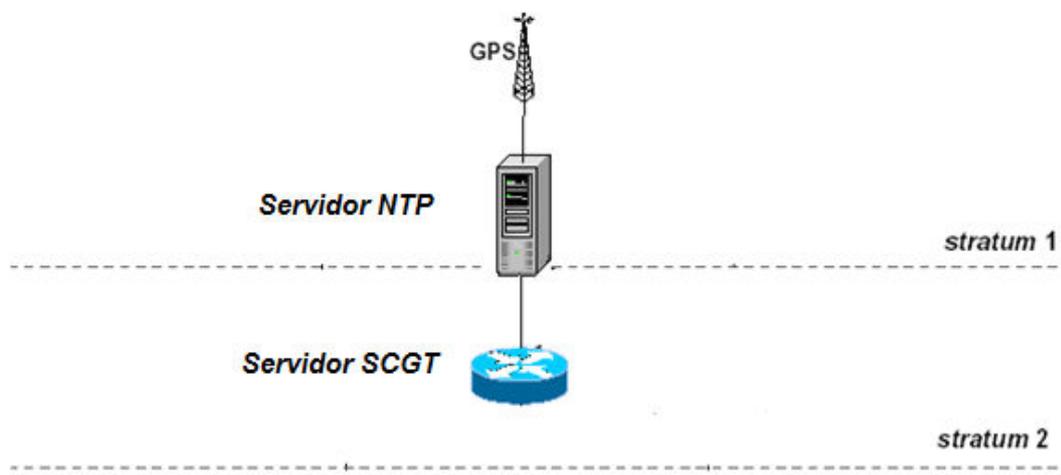
precisão conhecido, então se pode medir o intervalo entre dois eventos que ocorrem em diferentes computadores, recorrendo a seus relógios locais. Esta é a sincronização interna [COUL08].

6.7.1. Arquitetura Proposta

O SCGT pretende organizar, a partir dos seus pontos de presença, uma sincronização externa, com hierarquia de *stratum 2* do modo cliente-servidor utilizando o serviço do protocolo NTP descrito no capítulo 3 desta dissertação.

Veja na Figura 13, como está ligado o SCGT ao servidor NTP da rede GIGA:

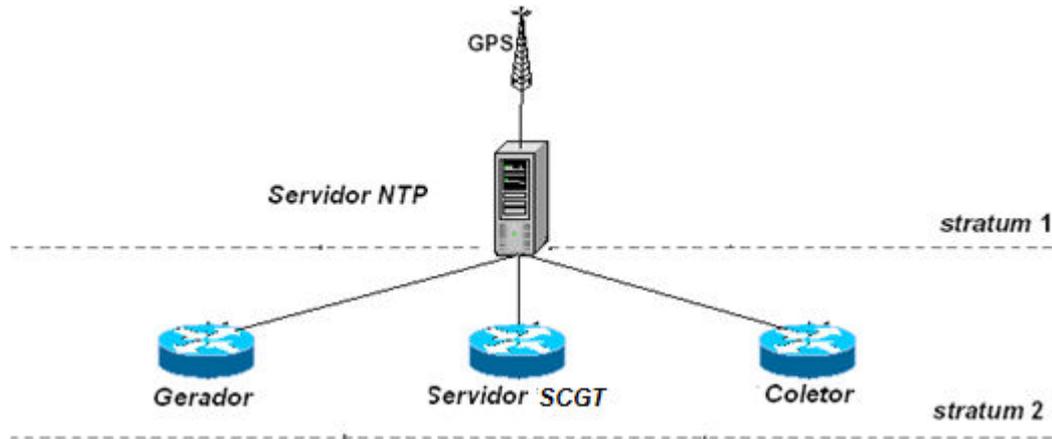
Figura 13: Sincronismo NTP na rede GIGA



O servidor NTP de *stratum 1* da rede GIGA atualmente utiliza a tecnologia GPS, que obtém o tempo diretamente de um grupo de satélites com uma precisão de 1 microssegundo.

A máquina em que se encontra o servidor SCGT está atualmente funcionando como *stratum 2*, e obtém o tempo diretamente do servidor NTP. Dessa forma, o SCGT propõe que as máquinas geradoras e medidoras utilizem o sincronismo NTP, funcionando como *stratum 2* no modo cliente-servidor, resultando em um serviço mais estável e confiável para o usuário final, já que a diferença entre os relógios das máquinas tende a ser da ordem de microssegundos. Veja na Figura 14 a seguir a arquitetura proposta para o uso do NTP na rede GIGA.

Figura 14: Arquitetura de Sincronismo Proposta para a rede GIGA



6.7.2. Problemas de Sincronismo nas Máquinas para tráfegos Simultâneos

Para analisar a sincronização com que cada fluxo de tráfego é enviado simultaneamente por uma rede amplamente distribuída através do SCGT, é preciso ter em mente possíveis alterações de comportamento de cada máquina. Dentre estas:

- O sincronismo dos relógios locais das máquinas geradoras, uma vez que o início do tráfego tem como referência o relógio destas.
- O tempo que cada máquina geradora demora em processar os pacotes para serem enviados através do uso das ferramentas (IPERF e DITG).
- O atraso na leitura e execução interna do *script* de geração do tráfego, já que uma vez que este *script* é executado em *background* na máquina geradora, um processo interno de contagem de tempo a cada segundo é inicializado para a verificação da hora local.

Experimentos relacionados ao sincronismo entre as máquinas são mais bem detalhados no capítulo 7 desta dissertação.

6.8. Conclusão

Este capítulo teve por objetivo a descrição do SCGT, um sistema que propõe avaliar o desempenho da rede permitindo gerar diferentes perfis de tráfego simultâneos. Este sistema foi criado como uma forma de contribuição para o ambiente de medição ativa em rede, pois, além de fornecer uma interface gráfica para ser usada através da *Web*, adiciona outras funcionalidades às ferramentas destinadas a este uso.

Os parâmetros de entrada para compor o perfil de tráfego são fornecidos nativamente por cada ferramenta de geração de tráfego sintético IPERF e DITG, e o SCGT integra a estas a forma

como estes são enviados (simultânea ou não) e o modo como o experimento é realizado (agendado ou instantâneo).

Além disso, o SCGT permite realizar experimentos, dispensando o conhecimento do sistema operacional *Unix* e até mesmo das ferramentas que compõem o sistema. Este sistema permite que um pesquisador execute e resgate os resultados de seus experimentos, sem a necessidade da instalação de nenhum outro *software* em sua estação de trabalho. Mais que isso, é possível operar o sistema de qualquer computador, e com qualquer sistema operacional.

Capítulo 7

Análise e Projeto do SCGT

7.1. Introdução

O rápido avanço da tecnologia requer, cada vez mais, formas mais rápidas e confiáveis para desenvolvimento de sistemas a fim de atender aos requisitos dos usuários. É, portanto, essencial a esses desenvolvedores utilizar técnicas de especificação que ofereçam consistência, concisão e clareza do sistema. Concomitante a isto, é ainda de suma importância que a técnica de especificação possua um conjunto de conceitos bem definidos, de modo a permitir a verificação da conformidade de implementações de acordo com as respectivas especificações.

O SCGT visa manipular e coordenar eventos de geração de tráfego sintético, criando um novo conceito de medição ativa para redes de grande porte. Neste sentido, uma técnica de especificação formal, fazendo uso de conceitos bem definidos é usada para representar as propriedades significativas do sistema. O uso de formalismo durante a especificação permite a eliminação de ambigüidades, inconsistências às quais apenas seriam detectadas mais tarde durante as fases de implementação e testes do ciclo de desenvolvimento do sistema.

Para documentar a especificação do SCGT foi utilizada a linguagem UML (*Unified Modeling Language*), gerando diagramas através da ferramenta *Jude Community 5.0.1* [JUDE08].

7.2. Visão Geral

O SCGT possui integrado em sua arquitetura uma interface *Web*, responsável por apresentar informações de configurações para compor o experimento e apresentar os resultados em tela, e o módulo de Coordenação, responsável por configurar o experimento e manipular os processos de geração de tráfego sintético, bem como extrair seus resultados.

7.3. Requisitos do Sistema

Visando solucionar os problemas decorrentes das dificuldades encontradas ao medir ativamente e simultaneamente o desempenho de uma ampla rede de telecomunicações, surgiu a necessidade de desenvolver uma solução de fácil operacionalidade, capaz de integrar ferramentas de geração de tráfego sintético, já existentes, e que pudesse ser feito de qualquer lugar de uma rede.

Diante deste escopo, foram levantados requisitos não-funcionais que o SCGT deve obedecer para se alcançar um resultado satisfatório. Estes requisitos são:

1. Possuir interface gráfica;
2. Operar sobre um *Web browser* (navegador na *Web*);
3. Integrar ferramentas de geração de tráfego sintético de código aberto.
4. Coordenar os processos das ferramentas de geração de tráfego sintético de acordo com cada experimento.
5. Conectar automaticamente nas máquinas envolvidas nos experimentos.
6. Extrair e manipular informações sobre o tráfego gerado.

O SCGT é indicado por oferecer condições necessárias para gerar tráfegos simultâneos e de diferentes modos, de forma a medir o desempenho de uma rede de alta velocidade. Deste modo, a arquitetura do sistema utiliza o protocolo HTTP para trafegar informações através do uso de um *Web Browser*, permitindo assim uma maior flexibilidade entre o usuário e o sistema, atendendo aos requisitos anteriormente listados.

Usuários

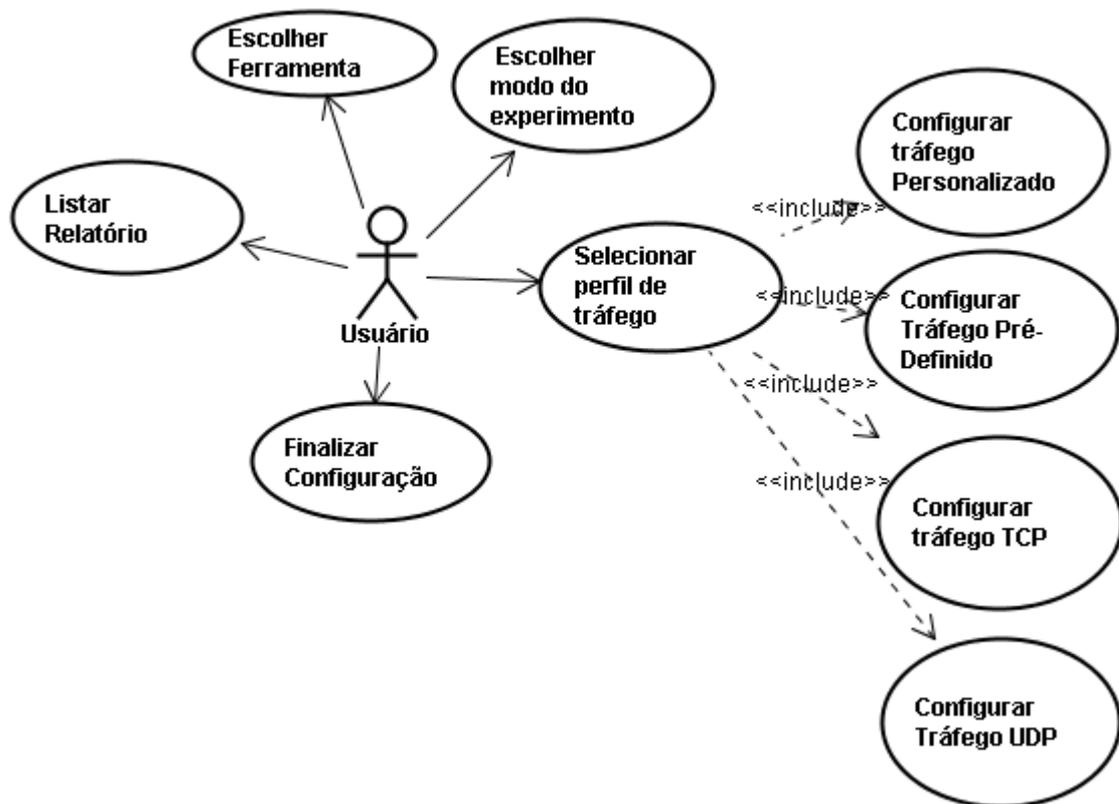
Os usuários inicialmente previstos para uso deste sistema são os pesquisadores do Projeto GIGA. Estes são responsáveis pela produção dos valores do perfil de tráfego a ser gerado. Existem diversos pesquisadores das mais diversas áreas, e a cada um deles pode ser atribuída a responsabilidade de um ou mais tráfegos gerados.

7.3.1. Diagrama de Casos de USO

O diagrama de casos de uso mostra os requisitos funcionais do sistema, dando uma idéia clara e consistente do que o sistema oferece. Este diagrama descreve as possíveis ações que um ator pode executar para alcançar determinado objetivo, mostrando as dependências entre as funcionalidades e os resultados esperados.

Para atender as funcionalidades apresentadas através dos requisitos do sistema, foram identificados 5 casos de uso. A Figura 15 a seguir ilustra estes casos de uso, identificando como o usuário final irá interagir com o SCGT.

Figura 15: Diagramas de Casos de Uso



1. Caso de Uso: Escolher Ferramenta

- **Ator:** Usuário
- **Objetivo:** Este caso de uso estabelece a interação entre o usuário e a ferramenta que irá gerar o tráfego através do SCGT.
- **Pré-Condição:** O usuário deve estar conectado à rede GIGA, e as máquinas que irão interagir nos experimentos devem atender aos pré-requisitos descritos no anexo D desta dissertação.
- **Pós-Condição:** A ferramenta é selecionada e armazenada no sistema.

- **Fluxo de Eventos:** O cliente seleciona a ferramenta que será utilizada no experimento. O Sistema registra a ferramenta.

2. Caso de Uso: Escolher modo do Experimento

- **Ator:** Usuário
- **Objetivo:** Este caso de uso estabelece a interação entre o usuário e o modo como cada ferramenta irá gerar o tráfego através do SCGT.
- **Pré-Condição:** A ferramenta deve ter sido escolhida.
- **Pós-Condição:** O modo do experimento da ferramenta é selecionado e armazenado no sistema.
- **Fluxo de Eventos:** O cliente seleciona o modo como será realizado o experimento (agendado ou instantâneo). O Sistema registra o modo do experimento da ferramenta.

3. Caso de Uso: Selecionar Perfil de Tráfego

- **Ator:** Usuário
- **Objetivo:** Selecionar o tipo de perfil de tráfego a ser gerado no experimento.
- **Pré-Condição:** O modo do experimento da ferramenta do sistema deve ter sido selecionado.
- **Pós-Condição:** Os parâmetros de entrada do tipo do perfil de tráfego são selecionados e armazenados no sistema.
- **Fluxo de Eventos:** O usuário seleciona o tipo de perfil de tráfego. O sistema, em função do perfil de tráfego e da ferramenta a utilizar no experimento (selecionada previamente), requisita ao usuário selecionar o tipo de tráfego.

Caso o usuário selecione UDP, INCLUI (*include*) Configurar Tráfego UDP.

Caso o usuário selecione TCP, INCLUI (*include*) Configurar Tráfego TCP.

Caso o usuário selecione Pré-Definido, INCLUI (*include*) Configurar Tráfego Pré-Definido

Caso o usuário selecione Personalizado, INCLUI (*include*) Configurar Tráfego Personalizado.

4. Caso de Uso: Configurar Tráfego UDP

- **Ator:** Usuário
- **Objetivo:** Configurar parâmetros de entrada para tráfego UDP.
- **Pré-Condição:** O usuário deve ter selecionado a ferramenta IPERF e o tipo de perfil de tráfego UDP.

- **Pós-Condição:** O experimento é configurado pelo sistema de acordo com os parâmetros de entrada preenchidos.
- **Fluxo de Eventos:** O sistema requisita o preenchimento dos parâmetros para tráfego UDP. O usuário deve preencher todos os campos obrigatórios de acordo com os parâmetros de entrada UDP fornecidos pelo sistema. O sistema registra os valores e configura o experimento a ser realizado.

A notar:

- ✓ Tempo de Duração - Deve ser inserido o valor do tempo em minutos
- ✓ Porta de Conexão com o servidor - Deve ser inserido o número da porta do servidor para conexão. Se não for preenchido nenhum valor, o número da porta assumido é 5001.
- ✓ Testar fluxo em paralelo – Deve ser marcada a opção desejada para o número de conexões paralelas para o mesmo sentido e porta de destino. Dentre elas, “Não” para a não realização de fluxos em paralelo, “2” (dois) para a realização de dois fluxos em paralelo, e “3” (três) para a realização de três fluxos em paralelo.
- ✓ Banda Máxima de Ocupação - Deve ser inserido o valor estabelecido para a largura de banda utilizando a unidade de dados bit/s. Se não for preenchido nenhum valor, esta é de 1 Mbit/s.
- ✓ Tamanho Máximo do Pacote UDP - Deve ser inserido o valor do tamanho máximo do pacote UDP, devendo incluir os cabeçalhos IP, Ethernet, mais informações de *payload*, utilizando a unidade de dados Bytes. Se não for preenchido nenhum valor, o tamanho do pacote é de 1470 Bytes.
- ✓ Relatório Final em Intervalo – Deve ser marcada a forma como deseja receber as estatísticas do relatório final. Dentre elas “Contínuo”, as estatísticas são apresentadas de segundo a segundo, e “Periódico”, as estatísticas são somadas e apresentadas de acordo com o tempo especificado na opção “Intervalo de Tempo”.
- ✓ Intervalo de Tempo – Esta opção somente é válida para “Relatório Final em intervalo Periódico”, onde deve ser colocado o intervalo de tempo em segundos, da forma como deseja receber as estatísticas do relatório final.
- ✓ Escolha a hora do envio – Esta opção somente é válida para tráfego no modo agendado. Deve ser colocado a hora no formato “00” e o minuto no formato “00” para o início de realização do(s) experimento(s).

5. Caso de Uso: Configurar Tráfego TCP

- **Ator:** Usuário

- **Objetivo:** Configurar parâmetros de entrada para tráfego TCP.
- **Pré-Condição:** O usuário deve ter selecionado a ferramenta IPERF e o tipo de perfil de tráfego TCP.
- **Pós-Condição:** O experimento é configurado pelo sistema, de acordo com os parâmetros de entrada preenchidos.
- **Fluxo de Eventos:** O sistema requisita o preenchimento dos parâmetros para tráfego TCP. O usuário deve preencher todos os campos obrigatórios de acordo com os parâmetros de entrada TCP fornecidos pelo sistema. O sistema registra os valores e configura o experimento a ser realizado.

A notar:

- ✓ Tamanho da janela TCP (Kbytes) - Deve ser inserido o valor do tamanho da janela TCP utilizando a unidade de dados Kbytes. Caso não seja inserido valor algum, é assumido como valor default 60 Kbyte/s.
- ✓ Tamanho máximo da MTU (Bytes) - Deve ser inserido o valor máximo para a unidade de transmissão TCP (MTU), utilizando a unidade de dados Bytes.
- ✓ Testar fluxo em paralelo – Deve ser marcada a opção desejada para o número de conexões paralelas para o mesmo sentido e porta de destino. Dentre elas, “Não” para a não realização de fluxos em paralelo, “2” (dois) para a realização de dois fluxos em paralelo e “3” (três) para a realização de três fluxos em paralelo.
- ✓ Tempo de Duração - Deve ser inserido o valor do tempo em minutos.
- ✓ Relatório Final em Intervalo – Deve ser marcada a forma como deseja receber as estatísticas do relatório final. Dentre elas “Contínuo”, as estáticas são apresentadas de segundo a segundo, e “Periódico”, as estatísticas são somadas e apresentadas de acordo com o tempo especificado na opção “Intervalo de Tempo”.
- ✓ Intervalo de Tempo – Esta opção somente é válida para “Relatório Final em intervalo Periódico”, onde deve ser colocado o intervalo de tempo em segundos, da forma como deseja receber as estatísticas do relatório final.
- ✓ Endereço IP de Origem e Destino – Deve ser colocado o endereço IP de origem, de onde será feito o envio do tráfego, e o endereço IP de destino, de onde será feita a coleta do tráfego. O preenchimento do segundo endereço IP de origem e destino é opcional, e caracteriza um tráfego simultâneo.
- ✓ Escolha a hora do envio – Esta opção somente é válida para tráfego no modo agendado. Deve ser colocado a hora no formato “00” e o minuto no formato “00” para o início de realização do(s) experimento(s).

6. Caso de Uso: Configurar Tráfego Personalizado

- **Ator:** Usuário
- **Objetivo:** Configurar parâmetros de entrada para tráfego personalizado.
- **Pré-Condição:** O usuário deve ter selecionado a ferramenta DITG e o tipo de perfil de tráfego personalizado.
- **Pós-Condição:** O experimento é configurado pelo sistema, de acordo com os parâmetros de entrada preenchidos.
- **Fluxo de Eventos:** O sistema requisita o preenchimento dos parâmetros para tráfego Personalizado. O usuário deve preencher todos os campos obrigatórios de acordo com os parâmetros de entrada fornecidos pelo sistema. O sistema registra os valores e configura o experimento a ser realizado.

A notar:

- ✓ Tipo de Métrica - Deve ser marcada a opção se deseja realizar testes utilizando a métrica OWDM (*One Way Delay Measurement*), ou RTTM (*Round Trip Time Measurement*). Se não for marcada nenhuma opção, a opção *default* será OWDM.
- ✓ Tipo de Distribuição do Tráfego - Deve ser marcada a opção de fazer distribuição de pacotes na forma Uniforme, Constante ou na forma Poisson.
- ✓ Valores para Distribuição Uniforme – Somente para o caso de escolha de distribuição uniforme, deve ser inserido o valor do tamanho do pacote mínimo e máximo, utilizando a unidade de dados Byte.
- ✓ Valores para Distribuição Constante – Somente para o caso de escolha de distribuição constante, deve ser inserido o valor do tamanho do pacote, devendo incluir os cabeçalhos IP, Ethernet, mais informações de payload, utilizando a unidade de dados Byte.
- ✓ Valores para Distribuição Poisson – Somente para o caso de escolha de distribuição Poisson, deve ser inserido o valor do tamanho médio do pacote, devendo incluir os cabeçalhos IP, Ethernet, mais informações de payload, utilizando a unidade de dados Byte.
- ✓ Duração dos Testes - Deve ser inserido o valor do tempo em minutos.
- ✓ Pacotes por segundo - Deve ser inserido o valor da quantidade de pacotes por segundo, que deseja enviar.
- ✓ Porta de Origem - Deve ser inserido o número da porta do cliente para conexão com o servidor.

- ✓ Porta de Destino - Deve ser inserido o número da porta do servidor para conexão com o cliente.
- ✓ Protocolo – Deve ser marcado o tipo de protocolo que deseja usar no experimento, sejam eles UDP, TCP ou ICMP.
- ✓ Escolha a hora do envio – Esta opção somente é válida para tráfego no modo agendado. Deve ser colocado a hora no formato “00” e o minuto no formato “00” para o início de realização do(s) experimento(s).

7. Caso de Uso: Configurar Tráfego Pré-Definido

- **Ator:** Usuário
- **Objetivo:** Configurar parâmetros de entrada para tráfego pré-definido.
- **Pré-Condição:** O usuário deve ter selecionado a ferramenta DITG e o tipo de perfil de tráfego pré-definido.
- **Pós-Condição:** O experimento é configurado pelo sistema, de acordo com os parâmetros de entrada preenchidos.
- **Fluxo de Eventos:** O sistema requisita o preenchimento dos parâmetros para tráfego Pré-Definido. O usuário deve preencher todos os campos obrigatórios de acordo com os parâmetros de entrada fornecidos pelo sistema. O sistema registra os valores e configura o experimento a ser realizado.

A notar:

- ✓ Duração dos Testes - Deve ser inserido o valor do tempo em minuto.
- ✓ Porta de Destino - Deve ser inserido o número da porta do servidor para conexão com o cliente.
- ✓ Tipo de Tráfego Pré-Definido – Deve ser marcado o tipo de tráfego pré-definido que deseja usar no teste, sejam eles DNS, Telnet, G.711.1-1 Amostra por pacote, G.711.1 - 2 Amostras por pacote, G.723.1, G.729.2 - 2 Amostras por pacote, G.729.3 - 3 Amostras por pacote, G.711.1 - 1 Amostra por pacote+VAD, G.711.1 - 2 Amostras por pacote+VAD, G.723.1+VAD, G.729.2 - 2 Amostras por pacote+VAD, G.729.3 - 3 Amostras por pacote+VAD.
- ✓ Escolha a hora do envio – Esta opção somente é válida para tráfego no modo agendado. Deve ser colocado a hora no formato “00” e o minuto no formato “00” para o início de realização do(s) experimento(s).

8. Caso de Uso: Finalizar Configuração.

- **Ator:** Usuário
- **Objetivo:** Finalizar configuração do experimento.
- **Pré-Condição:** Os parâmetros de entrada para o perfil de tráfego devem estar corretamente preenchidos.
- **Pós-Condição:** O experimento é inicializado pelo sistema.
- **Fluxo de Eventos:** O usuário marca a opção de “Gerar o tráfego” após preencher corretamente os parâmetros de entrada para o perfil do tráfego a ser gerado. O sistema inicializa os processos e executa-os de acordo com o modo a ser inicializado o experimento (instantâneo ou agendado).

9. Caso de Uso: Listar Relatório

- **Ator:** Usuário.
- **Objetivo:** Selecionar relatório através de uma listagem.
- **Pré-Condição:** O experimento de geração de tráfego deve ter terminado totalmente e o relatório deve estar no repositório de dados.
- **Pós-Condição:** É apresentado o relatório final selecionado.
- **Fluxo de Eventos:** O relatório final é buscado na base de dados. O sistema apresenta uma listagem com todos os relatórios recebidos. O usuário seleciona o relatório para visualização. O sistema registra o relatório selecionado apresentando-o em tela.

7.4. Diagrama de Classes

O diagrama de classes ilustrado na Figura 16 representa o relacionamneto das classes que compõem o SCGT, bem como os atributos e métodos anteriormente descritos.

A classe SCGT reúne todo procedimento envolvido na inicialização e finalização do tráfego configurado pelo usuário, além de criar instâncias para outras classes.

A classe Ferramenta reúne os procedimentos envolvidos na adição de uma ferramenta e no modo de funcionamento da mesma.

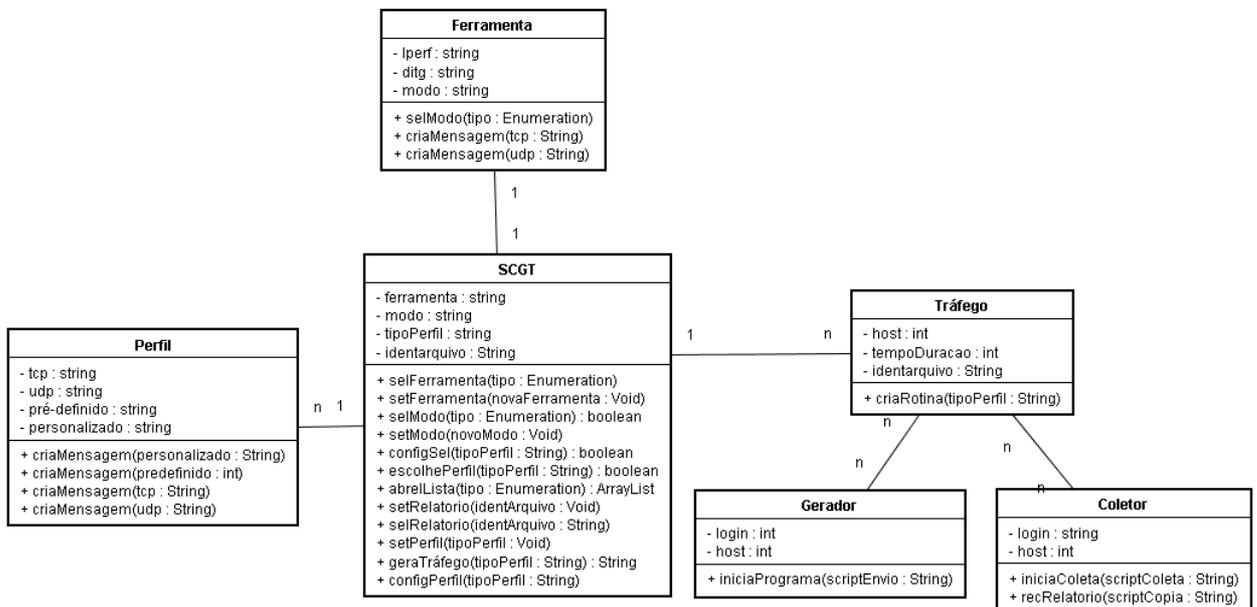
A classe Perfil representa os diferentes tipos de perfil atribuídos pelo usuário.

A classe Tráfego representa cada tipo de tráfego configurado pelo usuário e inicializado pelo sistema.

A classe Gerador representa as máquinas responsáveis por gerar cada perfil de tráfego configurado pelo usuário.

A classe Coletor é utilizada para representar as máquinas responsáveis por coletar o tráfego.

Figura 16: Diagramas de Classe



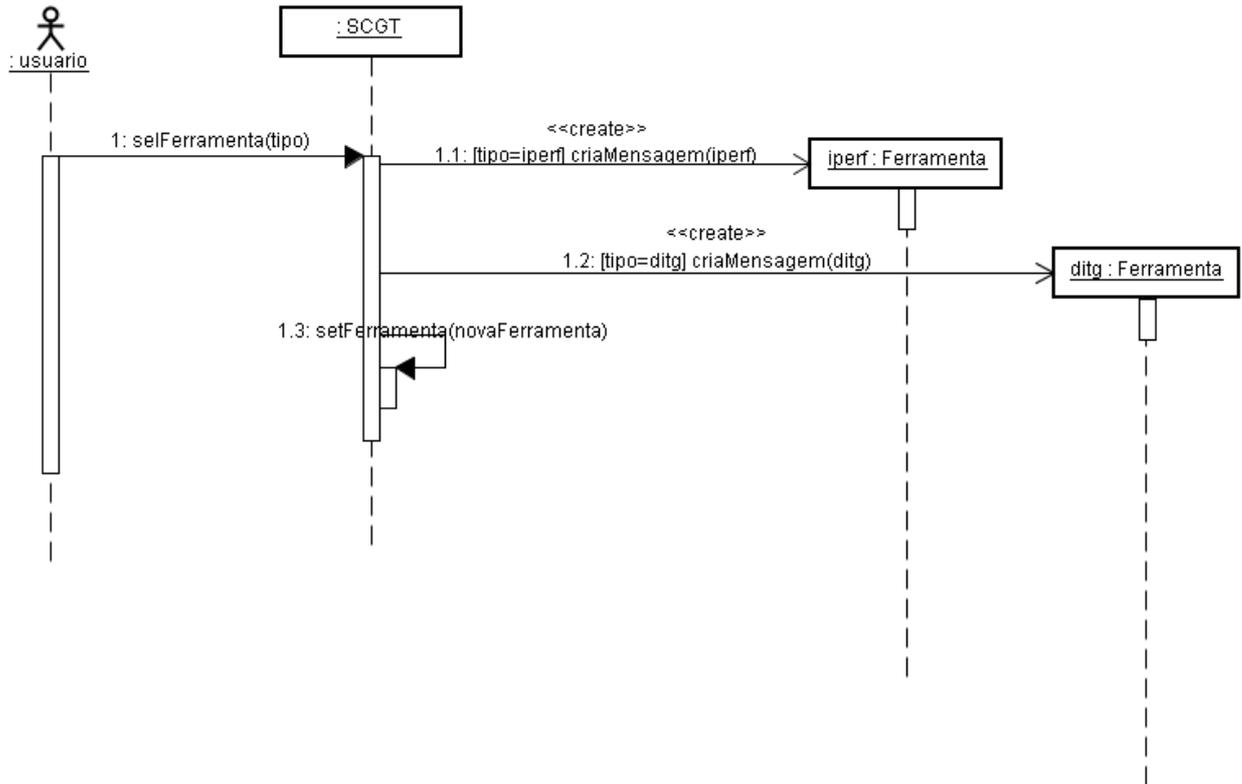
7.4.1. Diagramas de Seqüência

O diagrama de seqüência ilustrado nas Figuras 17, 18 e 19, 20, 21 e 22, mostram o fluxo de mensagens trocadas entre objetos para chegar a um objetivo em comum. Estes diagramas descrevem o comportamento de cada objeto, ao longo do tempo, durante a execução de uma funcionalidade (ou parte de uma funcionalidade) do SCGT. Em outras palavras, o diagrama de seqüência modela os processos definidos no diagrama de casos de uso, utilizando troca de mensagens através de objetos em um mesmo cenário.

1. Caso de uso Escolher Ferramenta

O diagrama de seqüência exibido na Figura 17 representa a seleção da ferramenta a ser utilizada pelo SCGT. O usuário executa o método “selFerramenta”, passando como parâmetro o tipo da ferramenta. A classe SCGT invoca a classe Ferramenta apresentando o tipo de ferramenta a ser configurada para o experimento.

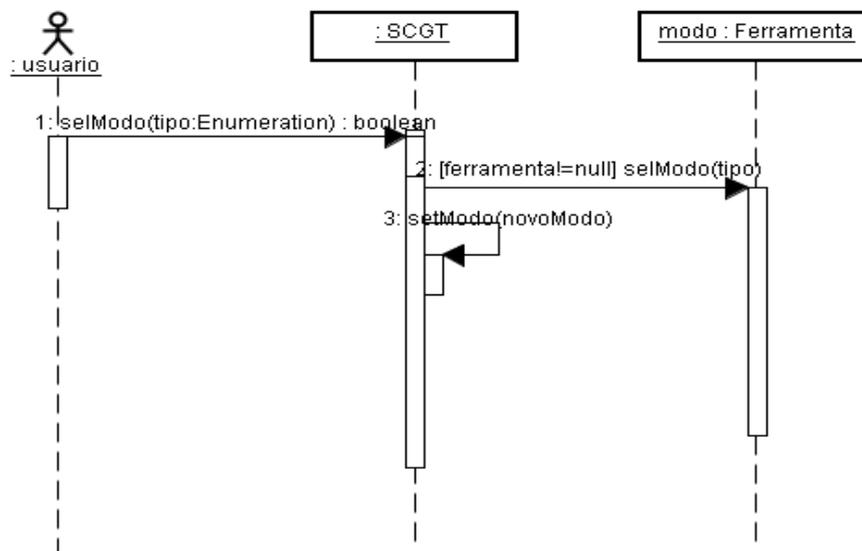
Figura 17: Diagrama de Seqüência-Ferramenta



2. Caso de uso Escolher Modo do Experimento

O diagrama de seqüência exibido na Figura 18 representa a seleção do modo do experimento a ser utilizado pelo SCGT. O usuário executa o método “selModo” passando como parâmetro o tipo do modo do experimento a ser utilizado. Caso a ferramenta tenha sido escolhida anteriormente, a classe SCGT instancia o objeto da classe Ferramenta disponibilizando o modo do experimento a ser utilizado.

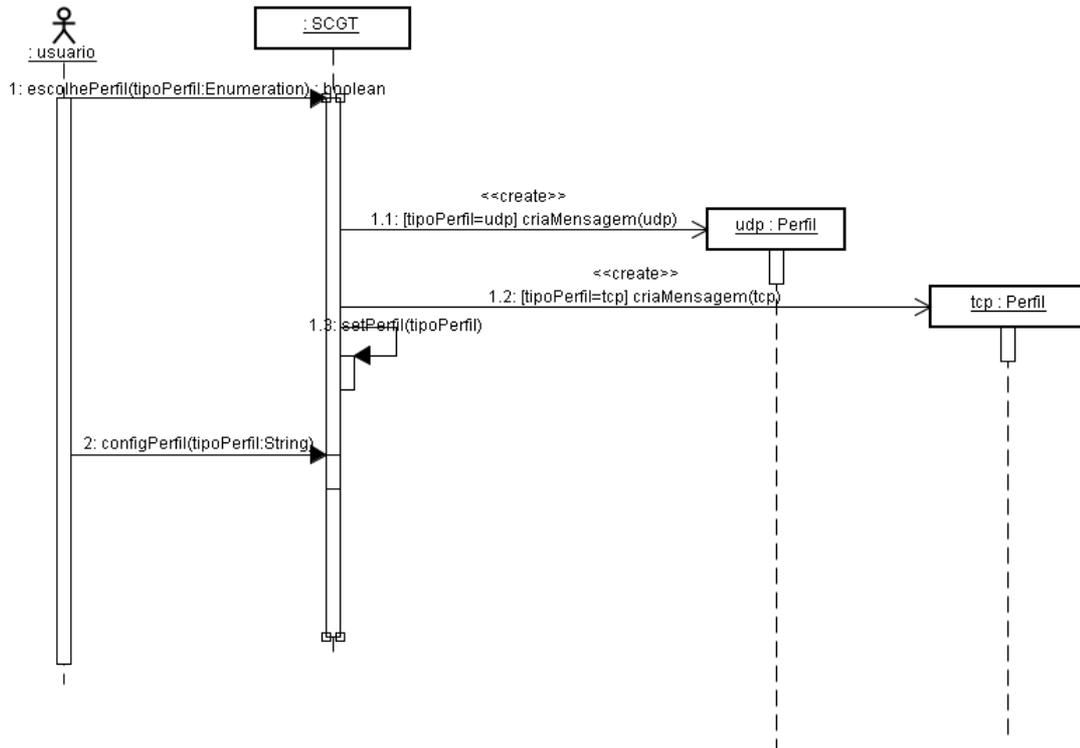
Figura 18: Diagrama de Seqüência-Modo



3. Caso de uso Selecionar Perfil de Tráfego para Ferramenta IPERF

O diagrama de seqüência exibido na Figura 19 representa a criação de um perfil de tráfego da ferramenta IPERF. O usuário solicita que um perfil de tráfego seja selecionado executando o método “selPerfil” e passando como parâmetro o tipo do perfil. Em seguida a classe SCGT instancia o objeto da classe Perfil de acordo com o tipo do perfil selecionado pelo usuário (UDP ou TCP), e o usuário configura o perfil de tráfego.

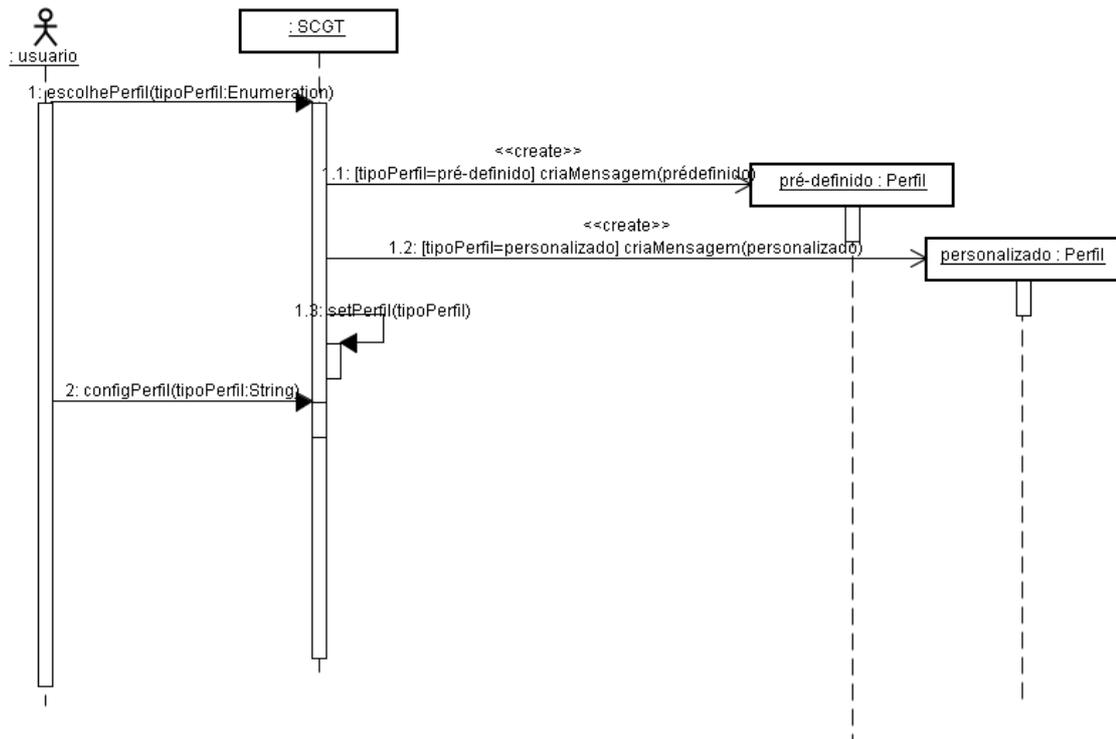
Figura 19: Diagrama de Seqüência-Perfil IPERF



4. Caso de uso Selecionar Perfil de Tráfego para Ferramenta DITG

O diagrama de seqüência exibido na Figura 20 representa a criação de um perfil de tráfego da ferramenta DITG. O usuário solicita que um perfil de tráfego seja selecionado executando o método “selPerfil” e passando como parâmetro o tipo do perfil. Em seguida a classe SCGT instancia o objeto da classe Perfil de acordo com o tipo do perfil selecionado pelo usuário (personalizado ou pré-definido), e o usuário configura o perfil de tráfego.

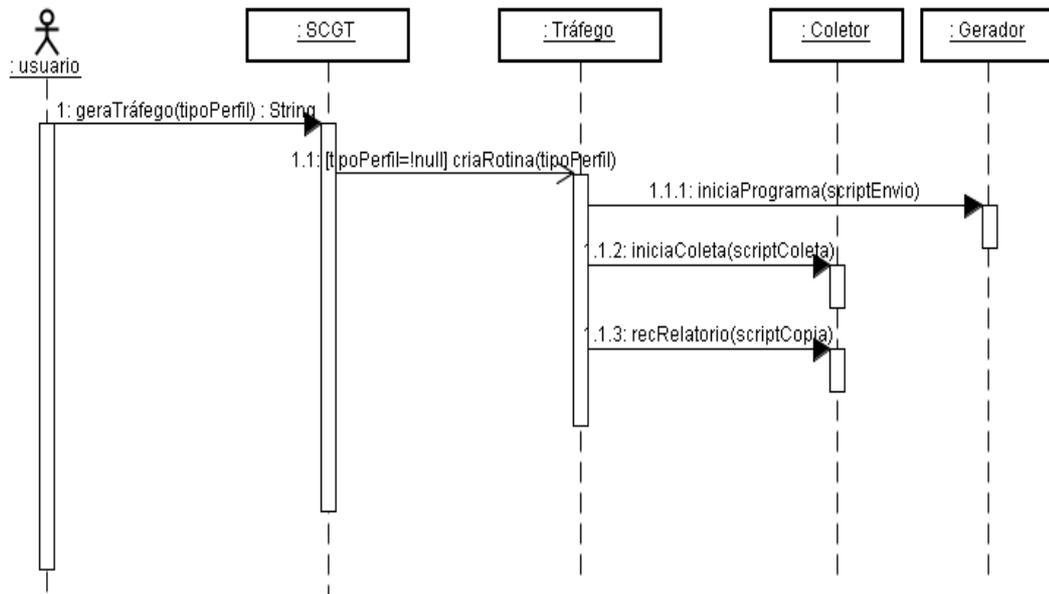
Figura 20: Diagrama de Seqüência-Perfil DITG



5. Caso de uso Finalizar Configuração

O diagrama de seqüência exibido na Figura 21 representa a finalização da configuração e o início do experimento de geração e coleta do tráfego. O usuário finaliza a configuração passando o parâmetro “geraTráfego”. Se os parâmetros de configuração do tipo de perfil foram preenchidos, a classe SCGT invoca a classe Tráfego a fim de criar as rotinas a serem executadas no experimento. Em seguida, a classe Tráfego invoca a classe Coletor e Gerador, inicializando e finalizando cada rotina do experimento.

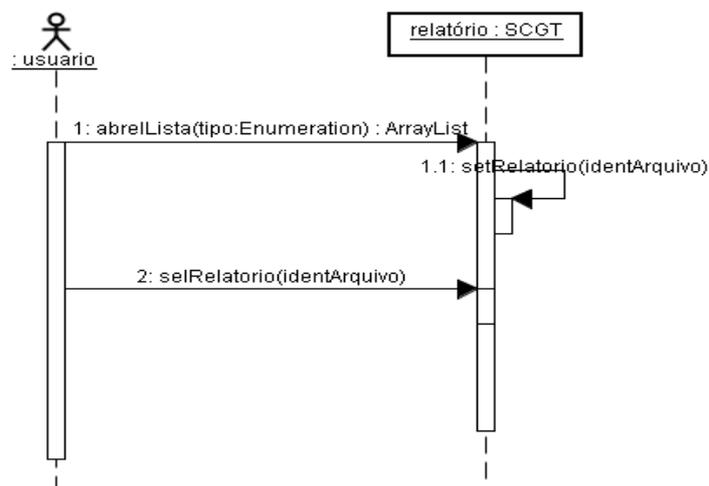
Figura 21: Diagrama de Seqüência-Finalizar



6. Caso de uso Listar Relatório

O diagrama de seqüência exibido na Figura 22 representa a apresentação e seleção do relatório feita pelo usuário. O usuário solicita a abertura da lista de relatórios. A classe SCGT busca todos os relatórios no repositório de dados de acordo com os atributos de identificação. Em seguida, o usuário seleciona o relatório passando como parâmetro o atributo de identificação.

Figura 22: Diagrama de Seqüência-Relatório



7.5. Conclusão

Este sistema foi projetado para ser um sistema modular, de modo a permitir a integração posterior de novos módulos/subsistemas ao já proposto. Dessa forma, a importância da fase de

análise e projeto do SCGT é considerada essencial para o entendimento da arquitetura envolvida por trás do sistema, e na organização das funcionalidades a serem criadas, facilitando a interação de novos módulos.

O diagrama de casos de uso mostrou os requisitos funcionais do sistema, dando uma idéia clara e consistente do que o SCGT pode oferecer. Foram mostradas através dos diagramas, as possíveis ações do sistema poderá executar para alcançar determinado objetivo, assim como as dependências das funcionalidades, e os resultados esperados.

Capítulo 8

Testes

8.1 Introdução

As diversas possibilidades de simulação de fontes de tráfego complexas, permitindo a repetição diversas vezes do mesmo padrão de tráfego e obtendo informações não apenas sobre os pacotes recebidos, mas também sobre os pacotes enviados, é uma característica altamente desejável em uma ferramenta de geração e medição de tráfego. Tal capacidade permite a realização de experimentos simulando eventos reais repetidamente, o que facilita a detecção e o diagnóstico de problemas na rede.

Cada vez mais as simulações devem refletir não apenas a grande escala dos cenários reais, mas também a enorme variedade das fontes de tráfego. Isto se refere em termos tanto de protocolos como de padrões de geração de dados, onde a simulação de tráfegos de diferentes camadas pode auxiliar na configuração de redes de alto desempenho.

O objetivo deste capítulo é avaliar o desempenho do SCGT quando este é integrado a rede GIGA. São feitos vários testes simulando diferentes perfis de tráfego, em um ambiente real. Dentre os diferentes tipos de dados possíveis de serem gerados e transmitidos, procurou-se utilizar uma mistura que reflita de forma realista o tráfego de uma rede IP operando em alta velocidade.

8.2. Topologia do SCGT

Atualmente estão integradas à rede GIGA três máquinas da Universidade Federal Fluminense (UFF), sendo duas separadas através de um enlace de fibra óptica com aproximadamente 1 quilômetro de distância. A máquina 1 e o servidor de NTP são compostos

fisicamente por servidores Blade da HP (*Hewlett Packard*), cujas configurações são detalhadas a seguir:

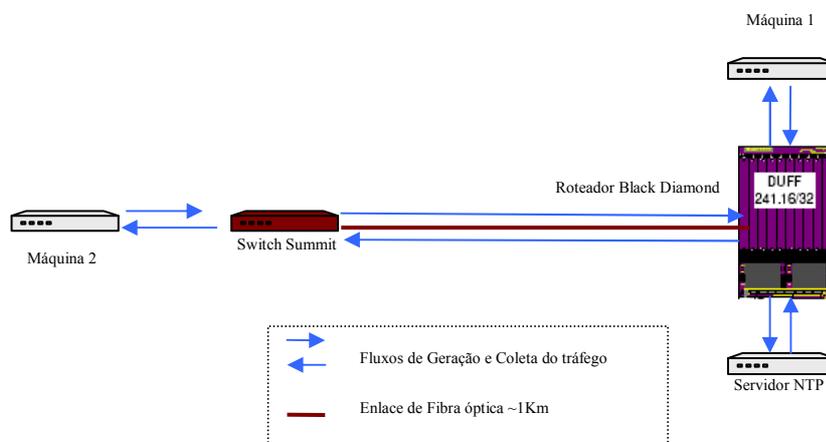
- Processador Intel Xeon 2.4 GHz;
- 1 GByte de Memória
- 1 HD de 80 GBytes
- 2 interfaces GigaEthernet UTP
- 3 interfaces GIGA ópticas

A máquina 2 é composta fisicamente por:

- Processador Intel Celeron 2.0 GHz;
- 512 MBytes de Memória
- 1 HD de 40 Gbytes;
- 3 Interfaces Ethernet UTP 10/100.

As entradas GIGA ópticas das máquinas anteriormente apresentadas são usadas para as ligações das estações de trabalho com o roteador *Black Diamond* conforme mostrado na Figura 23.

Figura 23: Topologia do SCGT na rede GIGA



Para subprojetos que usam a rede GIGA para transporte em alta velocidade, são configuradas VLAN's (Virtual LAN's) de interconexão entre os laboratórios.

Para os testes de geração de tráfego foram criadas duas VLAN's para as máquinas 1, 2 e o servidor de NTP. Dentre elas:

- Produção: Tem como objetivo a realização dos testes;
- Gerência: Tem como objetivo o acesso direto para configurações e instalações do SCGT e das ferramentas.

8.3. Descrição dos Testes

Estes testes se destinam a verificar o correto funcionamento dos SCGT, quando suas funcionalidades e características são testadas individualmente e paralelamente no ambiente de produção da rede GIGA, atribuindo uma situação de carga real.

Foram identificados os itens a serem testados, através das características de funcionamento da rede GIGA. A partir deste ponto foram definidas as métricas de interesse, de forma que possam fornecer a carga adequada para cada teste, e explicitar as informações relevantes mais facilmente.

Uma primeira etapa envolveu a geração e medição de tráfego utilizando cada uma das ferramentas individualmente e em um segundo momento é feito os testes simultâneos entre as ferramentas DITG e IPERF. Ao todo foram executados mais de trinta (30) testes, sendo que foram descritos neste capítulo apenas os de maior relevância.

Os itens a seguir descrevem a metodologia dos testes executados, e apresenta os resultados mais significativos. Os demais resultados são relatados nos apêndices ao final deste trabalho.

8.3.1 Primeira Etapa – Testes Individuais

Nos subitens a seguir, são apresentados os testes individuais utilizando o SCGT.

8.3.1.1. Taxa de Transferência máxima

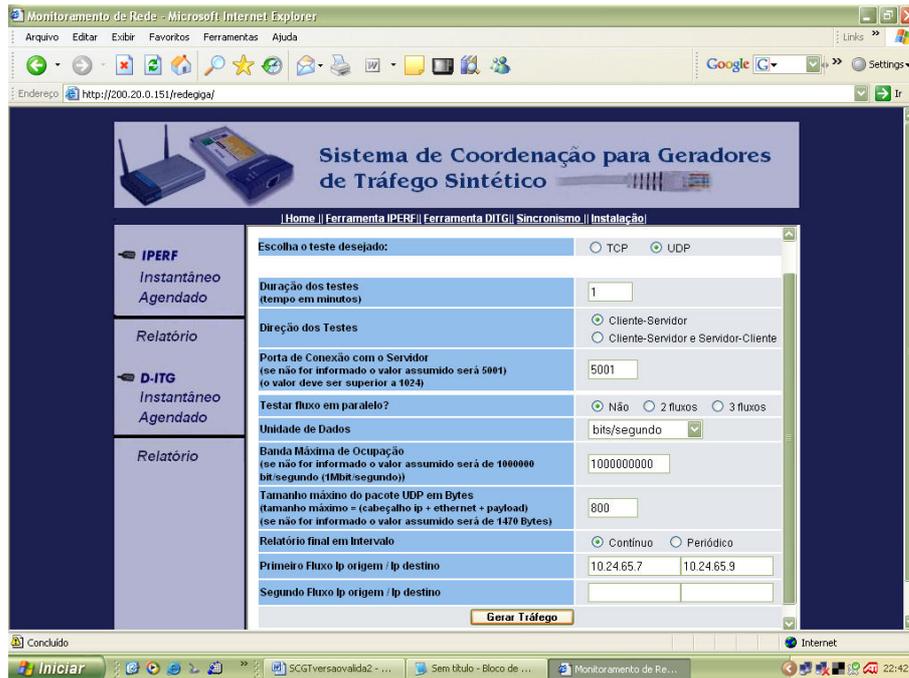
Os testes realizados envolvem a verificação experimental da taxa máxima de transmissão de dados entre as interfaces de rede da máquina 1 e o servidor NTP que suportam uma velocidade de até 1Gigabit/s. Os testes usam o protocolo de transporte UDP (*User Datagram Protocol*). O tráfego é gerado com a ferramenta IPERF utilizando larguras de banda de 1 Gigabit/s, com tamanho de pacotes de 200, 400, 600, 800 e 1000 Bytes.

Quadro 7: Descrição do teste de transferência

Teste	Ação	Resultado	Métrica de Interesse
Taxa Máxima da Ferramenta	Configuração de perfil de tráfego	Obter a maior taxa de transferência da ferramenta IPERF	Vazão

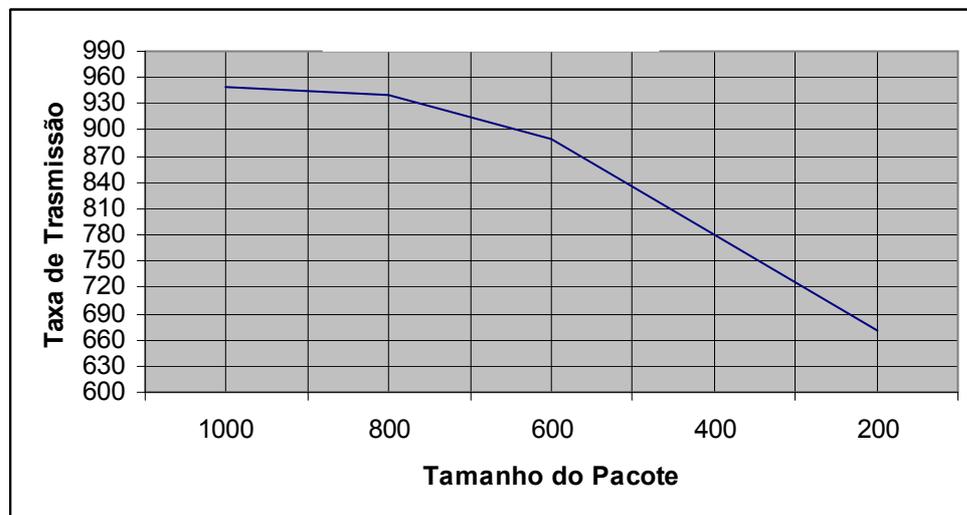
Como foram feitos vários testes, a seguir é mostrado na Figura 24 apenas o mais significativo, em que foi configurado um perfil de tráfego TCP no modo instantâneo.

Figura 24 –Taxa Gerada



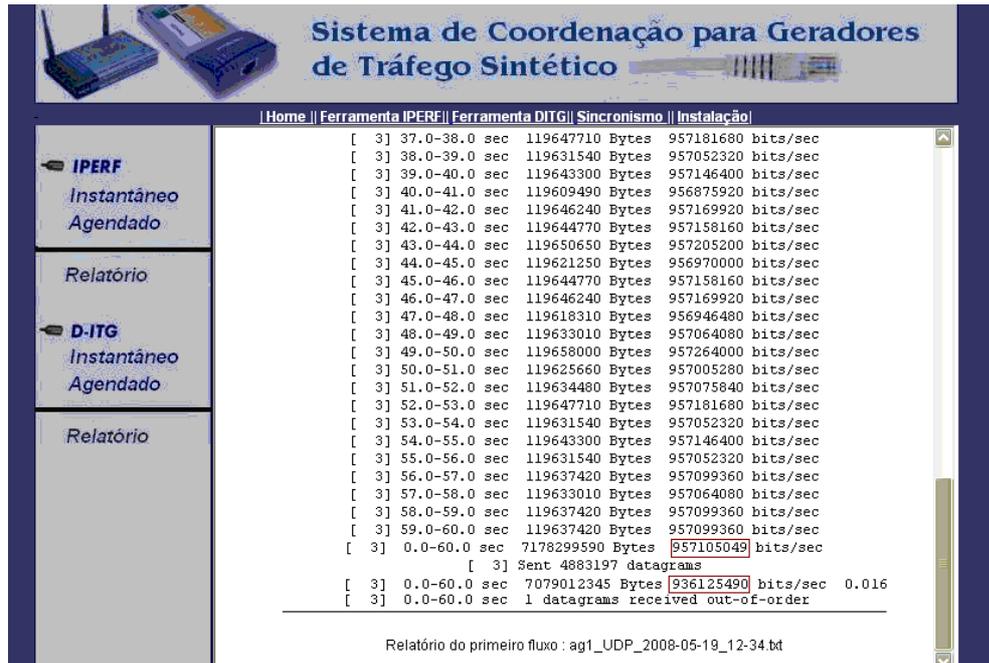
Resultado

Gráfico 1: Taxa de transmissão em função do tamanho do pacote



O Gráfico 1 obtido através da ferramenta *Excel*, mostra a taxa de transmissão dos dados em função do tamanho do pacote. Através deste, nota-se que a ferramenta IPERF consegue transmitir os dados em uma taxa próxima da largura de banda estabelecida a partir de um dado tamanho de pacote. Para pacotes com tamanho abaixo de 800 bytes, é possível notar que a diferença é bastante expressiva. Esta queda ocorre, porque é necessário um tempo mínimo para que cada pacote possa ser gerado. Veja a seguir, na Figura 25, o resultado do experimento configurado na Figura 24.

Figura 25: Taxa de Transferência



8.3.1.2. Aplicação DNS

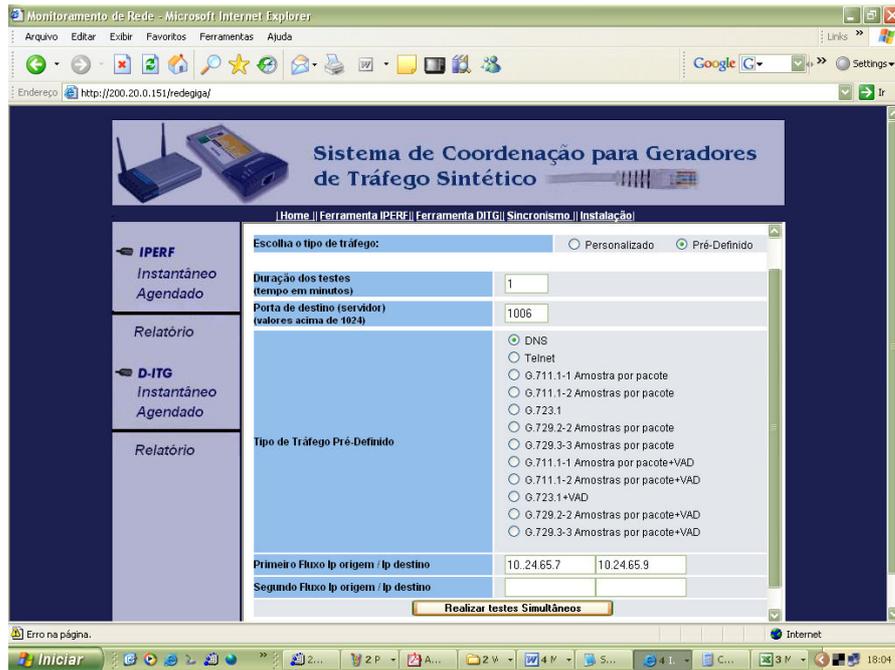
Este experimento foi realizado para testar a geração de tráfego para um perfil de uma aplicação DNS utilizando o perfil de tráfego personalizado da ferramenta DITG. Foi gerado através das máquinas 1 e 2 um único fluxo através do modo instantâneo do SCGT.

Quadro 8: Descrição do teste de DNS

Teste	Ação	Resultado	Métrica de Interesse
Simulação DNS	Configuração de tráfego para aplicação DNS	Obter medidas relacionadas a uma aplicação DNS	Perda de Pacotes Retardo <i>Jitter</i>

Foi feita a seguinte configuração descrita na Figura 26.

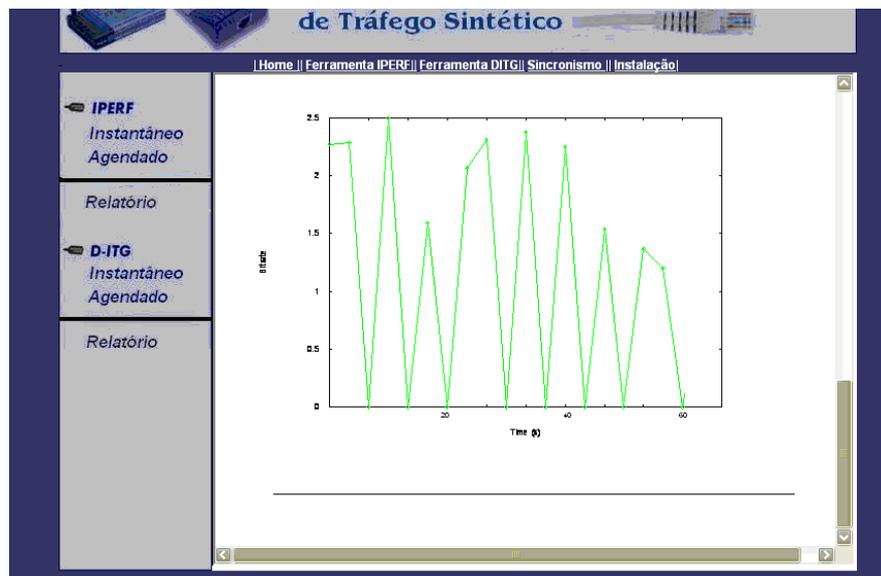
Figura 26: Perfil DNS



Resultado:

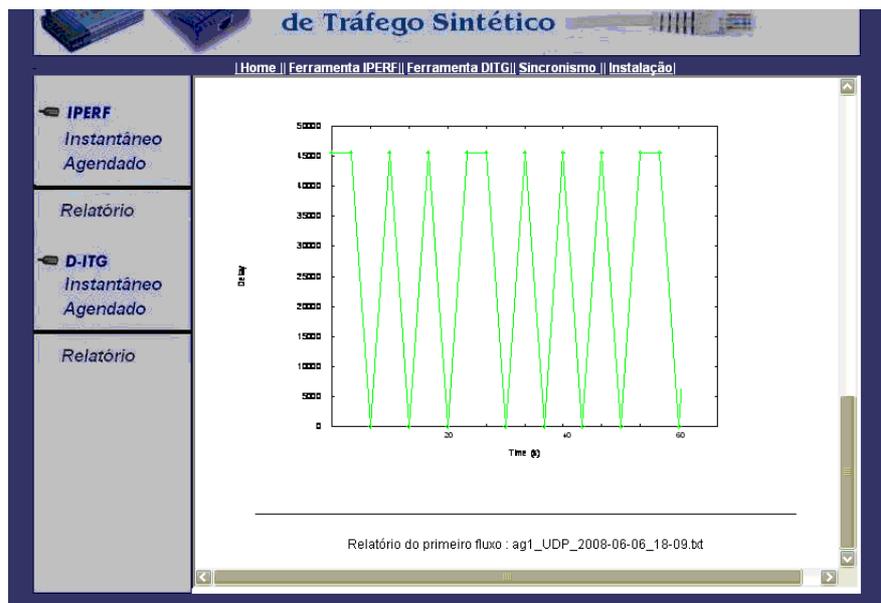
Através do Gráfico 2 é possível verificar períodos de transmissão de dados e períodos de silêncio aguardando o recebimento de pacotes de resposta e processando-os. Da mesma forma os valores máximo e médio estão dentro do esperado para uma aplicação DNS.

Gráfico 2: DNS- Bitrate x Time



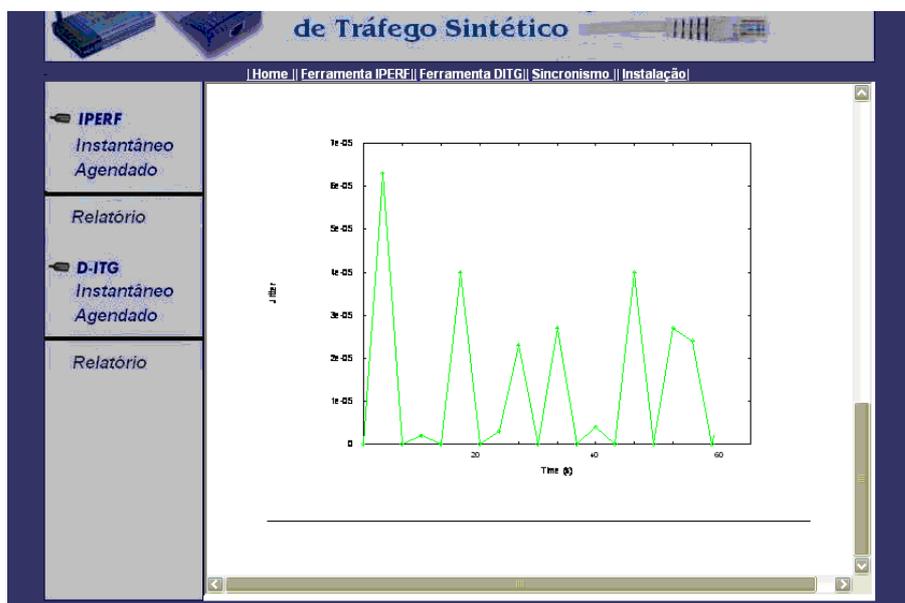
Acompanhando o padrão de transmissão, no Gráfico 3 verifica-se que o retardo cai a zero quando não há envio de pacotes.

Gráfico 3: DNS- Delay x Time



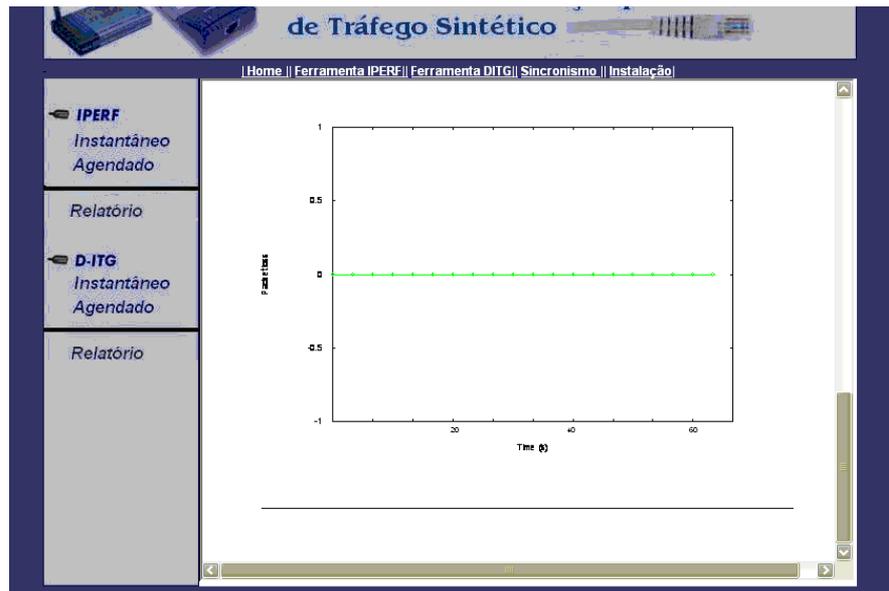
O Gráfico 4 apresenta grande variação do *jitter* também devido aos momentos de silêncio.

Gráfico 4: DNS- Jitter x Time



No Gráfico 5 não há perda de pacotes, pois os momentos de interrupção de envio de pacotes devem-se ao perfil do fluxo DNS e não a falhas ou perdas de qualquer espécie.

Gráfico 5: DNS- Packet x Time



8.3.2. Segunda Etapa - Testes Simultâneos

Nos subitens a seguir, são apresentados os testes simultâneos utilizando o SCGT.

8.3.2.1. Sincronismo entre as máquinas

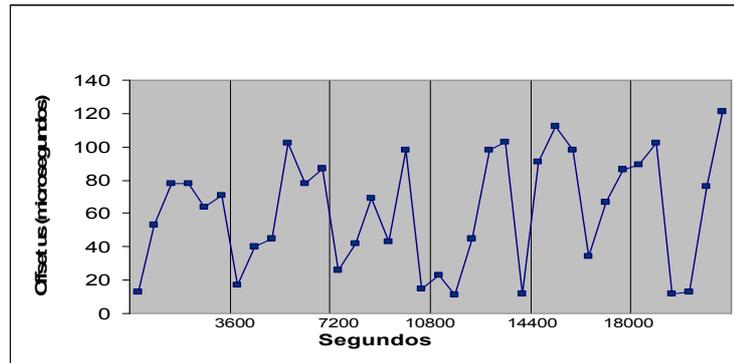
Seja cada máquina geradora cujo horário para o evento de envio do tráfego é informado pelo SCGT. Uma diferença considerável no relógio destas pode ser visto como uma falha geral do processo de envio do tráfego, uma vez que o SCGT toma como base o horário local disponibilizado pelas máquinas responsáveis pela geração do tráfego. Desta forma, o sincronismo das máquinas envolvidas no experimento, é um fator essencial para garantir o perfeito funcionamento do sistema.

O servidor de *stratum 1* (servidor de NTP) da rede GIGA é uma máquina representada pelo endereço IP 10.24.65.8 e é ligada a um relógio de referência que tem precisão de 1 μ s (microsegundo). As mensagens de pedidos de sincronização são enviadas periodicamente pelos clientes, podendo variar entre uma por minuto até uma a cada 17 minutos. O tempo que o servidor de NTP leva para sincronizar um relógio local para o último grau de precisão pode variar de alguns minutos até algumas horas, porém normalmente é em 5 minutos.

Como as máquinas 1 e 2 são configuradas por NTP através do mesmo servidor. Neste teste é observada a diferença dos valores de *offset* (diferença entre o relógio local e o de referência) das máquinas 1 e 2. Para compor este gráfico foi utilizado o comando “`ntpq -c pe`”, onde foi tirado uma amostragem a cada 10 minutos.

Resultado

Gráfico 6: Diferença dos valores de offset



Através do Gráfico 6, nota-se que a maior diferença de *offset* apresentada pelas máquinas 1 e 2 em relação ao mesmo relógio de referência ficou um pouco acima de 120 microssegundos, ou seja, essa é a maior diferença de sincronismo entre as máquinas, através do uso do NTP.

8.3.2.2. Conectividade

Para execução deste teste foram gerados 2 fluxos de tráfego simultâneos de cada ferramenta totalizando 4 fluxos, a fim de verificar a conectividade e o sincronismo obtidos entre as máquinas.

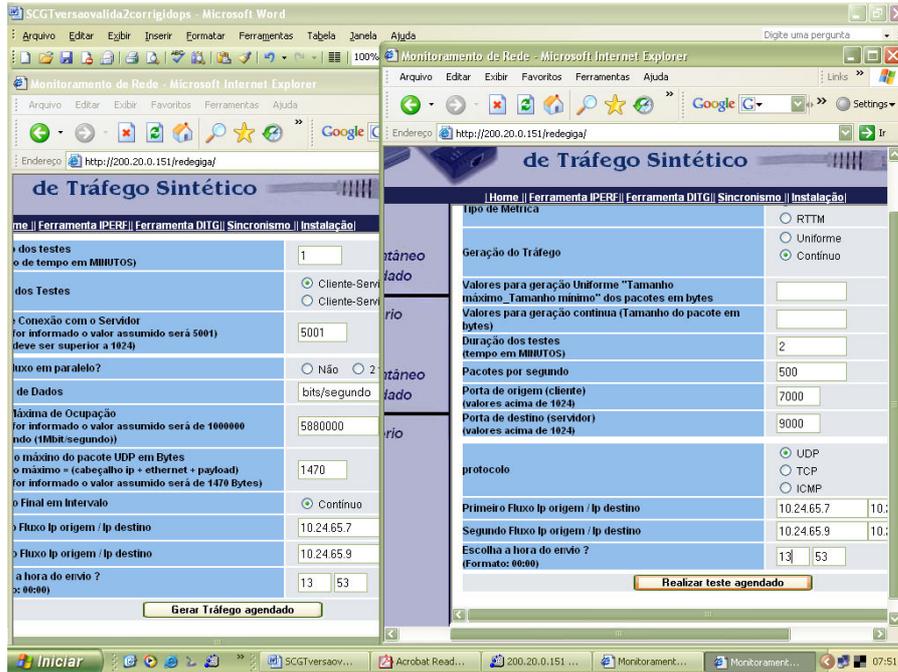
Este teste consiste em gerar o mesmo perfil de tráfego UDP para dois fluxos utilizando o mesmo sentido de medição entre as máquinas 1 e 2, através do uso das duas ferramentas IPERF e DITG. Estas vão iniciar o envio do tráfego simultaneamente a partir do modo agendado do SCGT.

Quadro 9: Descrição do teste de conectividade e simultaneidade

Teste	Ação	Resultado	Métrica de Interesse
Conectividade e Simultaneidade	Configuração de Perfil de tráfegos semelhantes	Conectividade entre as máquinas	Conectividade Retardo Perda de Pacotes

Com a ferramenta IPERF, foi gerado um fluxo UDP durante 1 minuto, e com a ferramenta DITG, foi gerado um tráfego com perfil semelhante durante 2 minutos, utilizando medição OWDM, para os mesmos sentidos de medição (endereço de origem e destino).

Figura 27: IPERF x DITG - Medição simultânea para mesmo sentido



Resultado

Nos Gráficos 8, 9 10, 11 e 12, os valores obtidos pela ferramenta DITG. São relatórios obtidos através do software *octave* utilizado pelo SCGT e mostram comportamento do tráfego no instante em que a ferramenta IPERF finaliza a geração do tráfego.

Gráfico 7: Simultâneo - *packet loss* x tempo – primeiro segundo

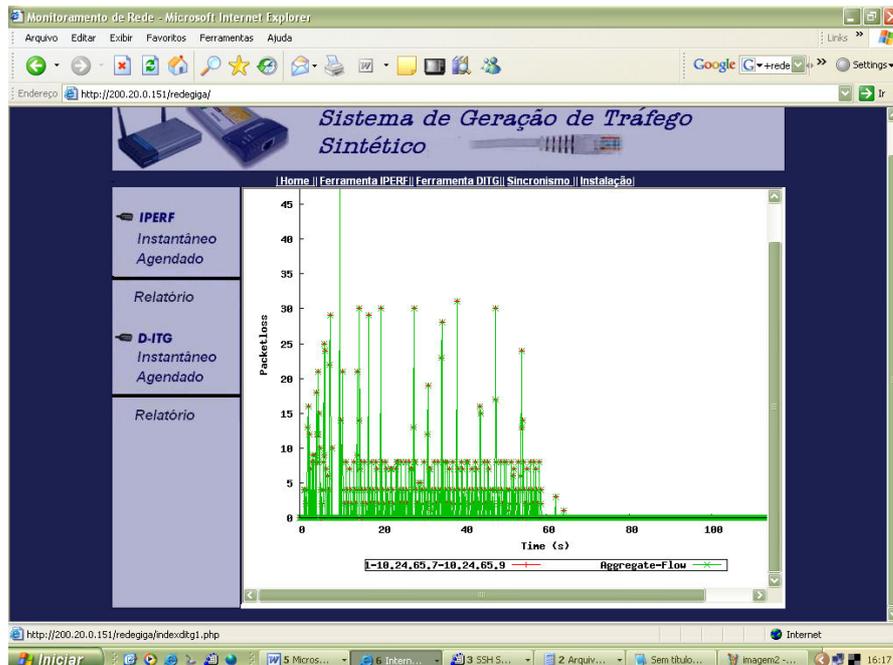
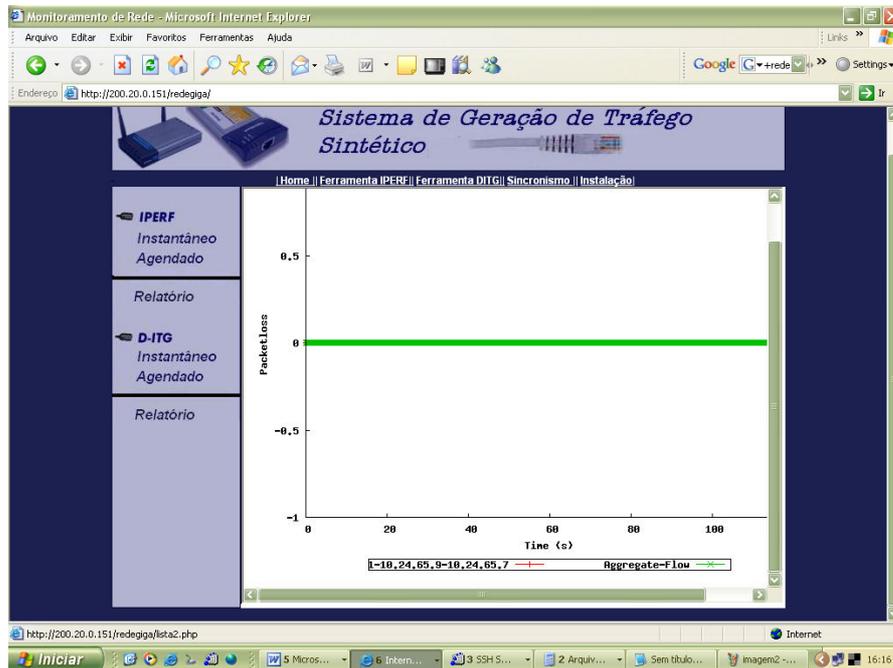


Gráfico 8: Simultâneo - packet loss x tempo – segundo sentido



No gráfico para o primeiro sentido, a perda de pacotes é influenciada pela execução simultânea do IPERF. No segundo sentido, não houve nenhuma influencia da ferramenta IPERF.

Gráfico 9: Simultâneo - delay x tempo – primeiro sentido

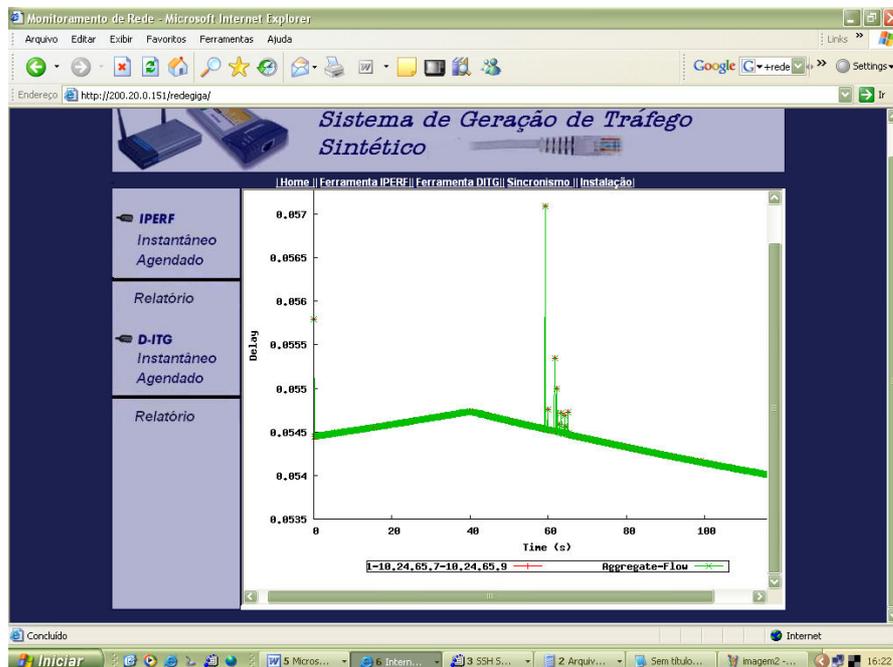
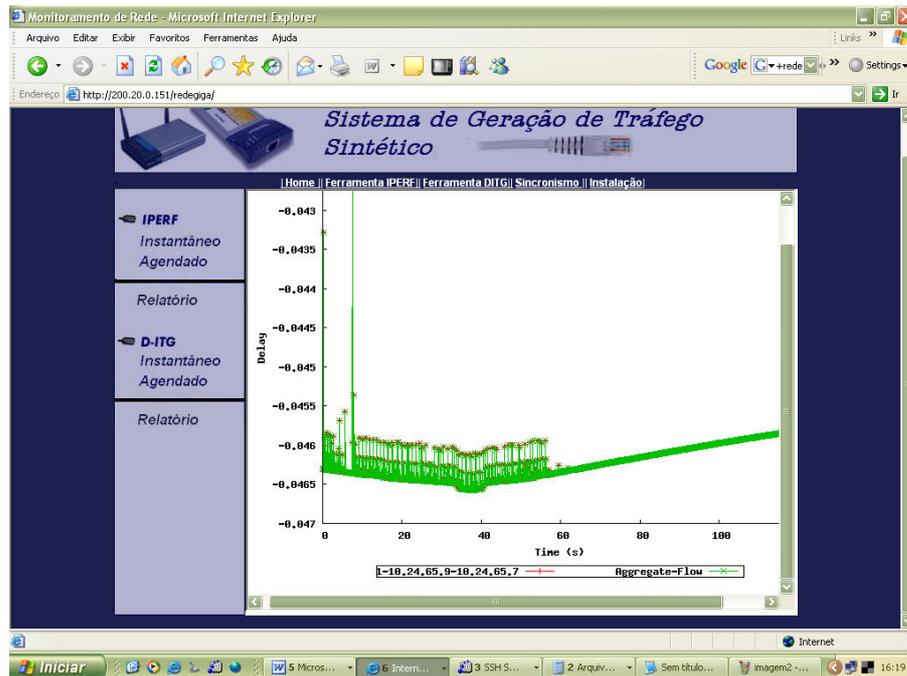


Gráfico 10: Simultâneo - *delay* x tempo – segundo sentido

Em ambos os sentidos, o retardo foi influenciado pela execução simultânea do IPERF, e pela sua interrupção.

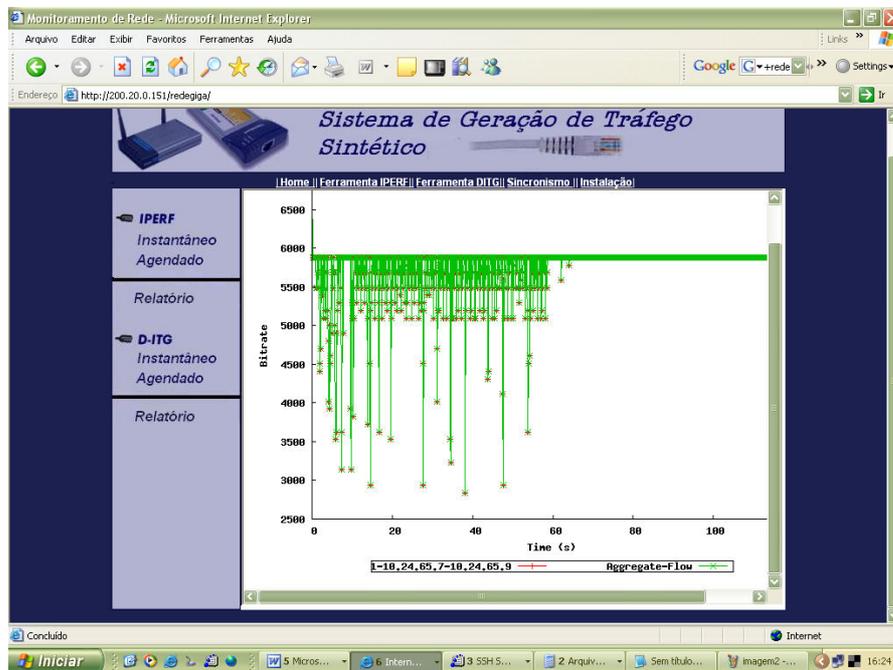
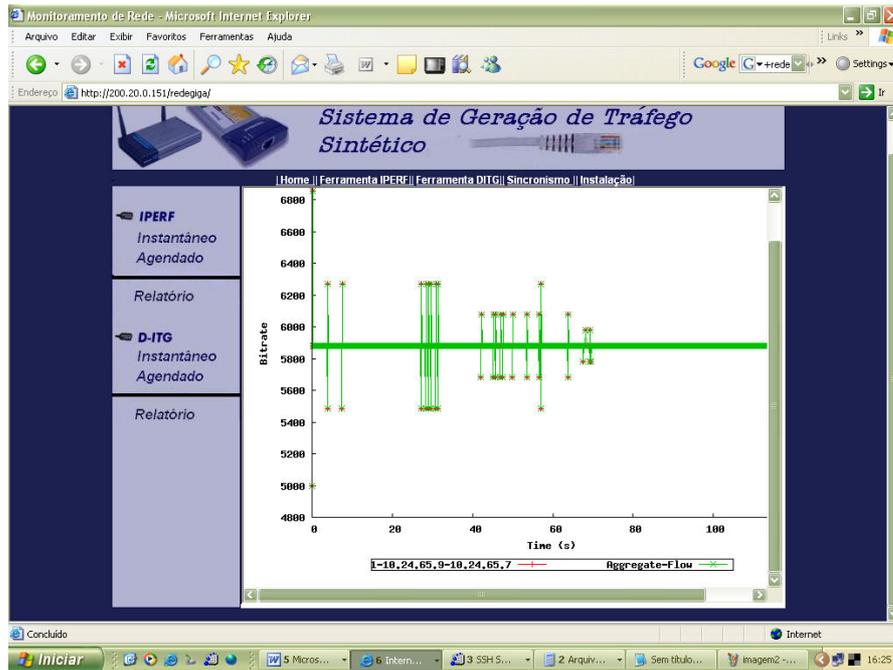
Gráfico 11: Simultâneo - *Bitrate* x tempo – primeiro sentido

Gráfico 12: Simultâneo - *Bitrate x tempo* – segundo sentido

Os valores da taxa de bit na chegada ficaram na média de 5,9 Mbit/s para os dois fluxos. No entanto, o primeiro fluxo parece ter sofrido mais influencia da ferramenta IPERF, do que o segundo fluxo.

8.3.2.3. Medidas Unidirecionais

Este teste consiste em validar medidas unidirecionais através da geração simultânea de pacotes UDP.

As medidas unidirecionais são consideradas de bastante relevância para aplicações que requerem maior exigência de desempenho da rede em um único sentido, como, por exemplo, aplicações multimídias, onde pacotes de vídeo com atraso menor do que 150 ms é aceitável, ao passo que para o áudio este valor seria de 400 ms.

Para medições unidirecionais é necessário que origem e destino estejam bem sincronizados, e para tal foi utilizado como máquina coletora o servidor NTP de *stratum 1* da rede GIGA. A escolha desta máquina para compor o teste, foi para evitar possíveis erros nas marcas de tempo (*timestamp*) dos pacotes no momento em que são coletados. O uso somente da ferramenta IPERF para este teste, justifica-se pelo fato de que o servidor NTP funciona com o sistema operacional Unix *FreeBSD* e a ferramenta DITG não permite instalação neste tipo de distribuição.

Os testes unidirecionais são baseados nas informações constantes da RFC que descreve o protocolo IP [RFC791] “dado um protocolo e um mesmo par origem-destino, a unicidade do

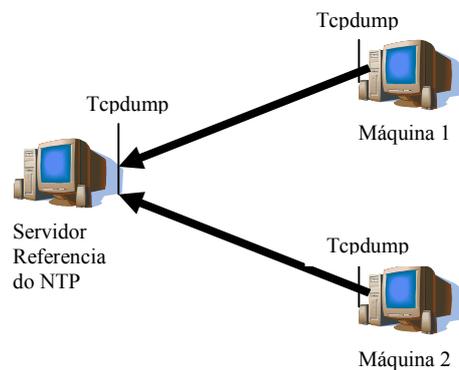
valor presente através do campo de identificação do cabeçalho IP prevalece enquanto o datagrama permanecer ativo na internet, sendo utilizado como identificação dos pacotes”. No quadro 9 é descrito as medidas e os resultados observados no teste unidirecional.

Quadro 10: Descrição do teste Unidirecional

Teste	Ação	Resultado	Métrica de Interesse
Medidas unidirecionais e Simultaneidade	Diferentes endereços de origem e mesmo endereço de destino	Medidas Unidirecionais e precisão da geração do tráfego simultâneo	Atraso Unidirecional Perda de Pacote

Para este teste, o primeiro passo foi utilizar uma ferramenta de captura nas máquinas destinadas a medição e geração do tráfego filtrando somente pacotes UDP originados dos dois fluxos (10.24.65.7 para o 10.24.65.8 e do 10.24.65.9 para o 10.24.65.8). Para tal foi utilizada a ferramenta *tcpdump*. Esta ferramenta é baseada na biblioteca libcap do *Unix*, e é destinada a capturar os pacotes trafegados em rede de telecomunicações. A figura 28 representa a configuração física das máquinas envolvidas no experimento.

Figura 28: Configuração para teste unidirecional



O segundo passo foi determinar o perfil do tráfego através do SCGT, onde foi atribuído um teste de 1 minuto de dois fluxos simultâneos no modo agendado utilizando um perfil de tráfego UDP, com configuração padrão (*default*) com pacotes de 1470 bytes e largura de banda de 1 Mbit/s através da ferramenta IPERF. Ambas as máquinas (1 e 2) estão enviando um fluxo para a máquina servidora do NTP que atua como *stratum 1* na rede GIGA e o horário marcado para início da geração dos tráfegos é para as 15:57 horas através do SCGT.

Resultado

A primeira verificação do teste foi o atraso unidirecional provocado por cada pacote na rede, bem como do sincronismo no envio do tráfego entre as máquinas. Como foram gerados mais de 50000 pacotes em cada fluxo, somente foi retirada uma amostragem dos pacotes dos

primeiros 10 segundos de transmissão. O atraso foi obtido percorrendo sequencialmente cada fluxo das máquinas 1 e 2 através das informações obtidas pelo *tcpdump*. Para cada datagrama capturado na máquina geradora foi buscado o seu equivalente na máquina medidora (com mesmo identificador). O primeiro elemento do par representa o tempo marcado pelo relógio das máquinas 1 e 2 e o segundo representa o tempo marcado (*timestamp*) pela máquina medidora no momento da coleta de cada pacote separadamente. O cálculo do retardo de cada pacote é obtido subtraindo-se o segundo elemento do par pelo primeiro. Nos Gráficos 13 e 14 é ilustrado o retardo provocado por cada pacote através dos primeiros 10 segundos de transmissão de cada fluxo, e o momento em que o primeiro pacote de cada fluxo é gerado.

Gráfico 13: Retardo Unidirecional Máquina 1 x Servidor NTP

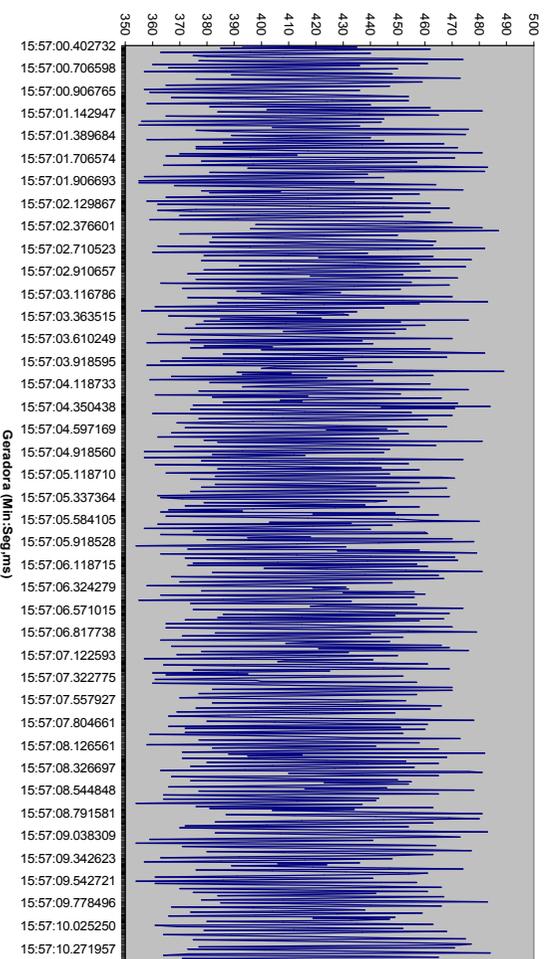
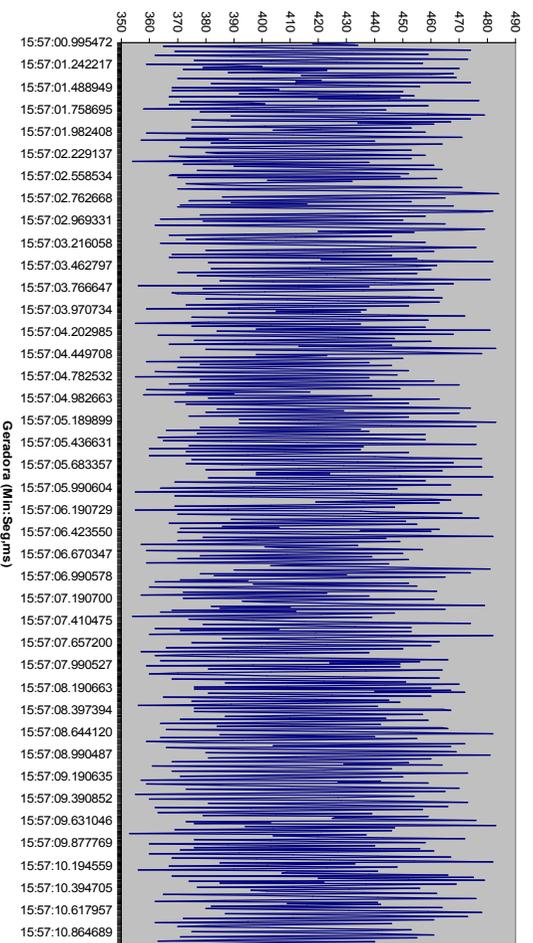


Gráfico 14: Retardo Unidirecional Máquina 2 x Servidor NTP



Através dos Gráficos 13 e 14 pode-se observar que os retardos se encontram dentro de uma faixa de tempo, o que caracteriza que os relógios das máquinas geradoras estiveram

sincronizados em relação a máquina medidora (servidor NTP) ao longo do experimento. Da mesma forma pode-se observar que:

- A máquina 1 enviou o primeiro pacote às 15:57:00.402.732 ms.
- A máquina 2 enviou o primeiro pacote às 15:57:00.995.472 ms.

A diferença do início de geração do tráfego entre as duas máquinas foi de 592.740 ms através do uso de SCGT. Ou seja, a máquina 2 atrasou o início da geração do tráfego 592.740 milissegundos em relação a máquina 1. Esta pequena diferença pode-se justificar pelo tempo de processamento dos pacotes e pela leitura e execução do script de geração de tráfego, podendo este ser prejudicado pelas limitações de velocidade causadas pelo hardware.

A segunda verificação no teste foi em relação a perda dos pacotes:

- No fluxo 1 não foi encontrado o ID de 3 pacotes na máquina receptora, ou seja, ocorreram 3 perdas de pacotes. Esta é expressa por uma porcentagem. Para este cálculo, basta subtrair os pacotes recebidos da quantidade de pacotes transmitidos e dividir pelo total de pacotes transmitidos.
- No fluxo 2 não foi detectado nenhuma perda de pacotes, uma vez que todos os pares de identificação foram encontrados até o final do teste.

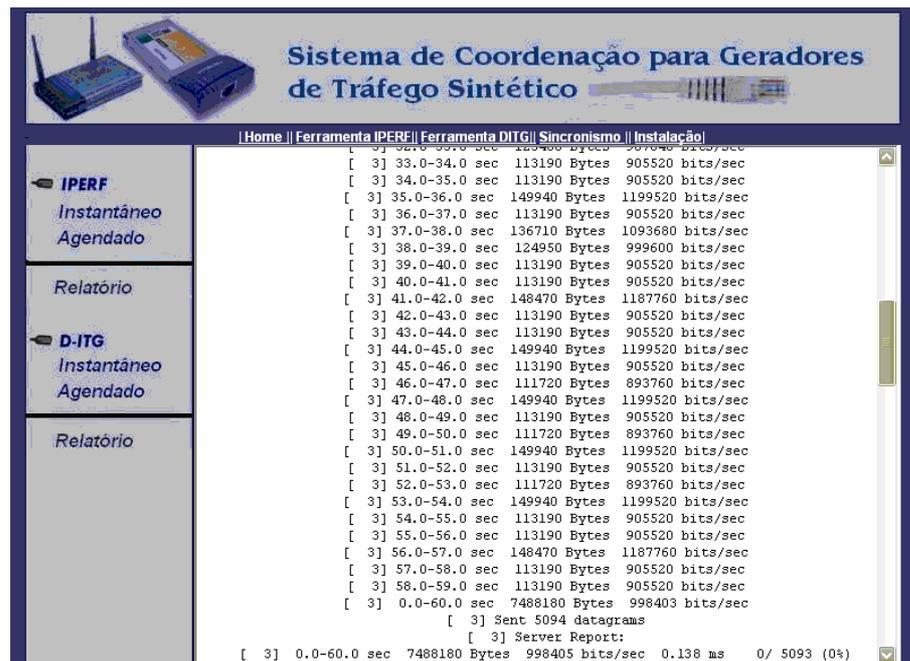
Nas Figuras 26 e 27, as estatísticas coletadas pela ferramenta IPERF ao longo de 1 minuto de experimento. Note através das Figuras 29 e 30 que os valores apresentados estão dentro do esperado pelo teste.

- Bytes Recebidos 7497000 bytes x 8 = 59976000 bit/s / Tamanho dos pacotes = 1470 x 8 11760 = 59976000 / 11760 = Total de pacotes na chegada = 5100 pacotes.
- Bytes Recebidos 7488180 bytes x 8 = 59905440 bit/s / Tamanho dos pacotes = 1470 x 8 11760 = 59905440 / 11760 = Total de pacotes na chegada = 5094 pacotes.

Figura 29: Resultado do primeiro fluxo



Figura 30: Resultado do Segundo fluxo



8.4. Conclusão

O SCGT é uma interface gráfica desenvolvida para ambiente *Web*, que incrementa funcionalidades às duas ferramentas de geração de tráfego de código aberto, IPERF e DITG. Dessa forma, vários testes são agendados e disparados simultaneamente a partir de qualquer sistema operacional, e de qualquer navegador *web*.

Apesar do SCGT ter sido pensado originalmente para experimentos relacionados ao desempenho de uma rede de alta velocidade, este pode ser utilizado em qualquer tipo de rede seja no acesso ou no núcleo, indo do Frame Relay ou ATM, passando pelo ADSL até o PPP. Isto faz com que o mesmo seja um sistema muito útil no diagnóstico de redes através de medições ativas.

Através deste capítulo, pode-se observar que as duas ferramentas que compõem o SCGT apresentaram desempenho satisfatório, atendendo às suas especificações e gerando tráfego de acordo com o previsto. Assim, é possível concluir que as funcionalidades agregadas ao SCGT quando em uso com as ferramentas, se faz extremamente útil, apresentando características que facilitaram a obtenção de informações nos testes realizados. Entretanto, este sistema ainda precisa evoluir em alguns aspectos, possibilitando uma maior integração com outros recursos, e ferramentas de geração de tráfego sintético.

Capítulo 9

Conclusão e Trabalhos Futuros

9.1. Conclusão e Considerações Finais

O desenvolvimento de aplicações *Web* tem se tornado cada vez mais freqüente pelos benefícios que esta arquitetura oferece como independência da plataforma e da localização da máquina, bastando apenas que esta esteja conectada à *Internet*. Se junta a este fator, o constante avanço tecnológico, fazendo com que as redes corporativas fiquem cada vez maiores e complexas, tornando inviável o acesso a determinados tipos de serviços de medição ativa em redes de dados.

E é exatamente para o contexto descrito que foi desenvolvido o SCGT (Sistema de coordenação para Geradores de Tráfego). Este sistema possui uma proposta inovadora de medir ativamente os diferentes pontos de uma rede, possibilitando disparar tráfegos de forma simultânea, a qualquer hora e de qualquer lugar, utilizando a arquitetura *Web* como referência. Além disso, o módulo SCGT responsável pela interação e coordenação das ferramentas de geração de tráfego sintético foi todo constituído em código aberto utilizando a linguagem PHP [PHP08], e está totalmente descrito no apêndice B desta dissertação, possibilitando que novas ferramentas de geração de tráfego sintético sejam integradas a este sistema.

O SCGT não está totalmente difundido através da rede GIGA, o que tornou a sua utilização um pouco limitada, pois aspectos relacionados ao sincronismo utilizados entre as máquinas precisam ser mais bem explorados, utilizando vários servidores NTP de *stratum* 1 atuando dinamicamente na rede. Contudo, durante o desenvolvimento desta dissertação, foi possível observar o enorme potencial contido neste sistema, devido ao fato deste permitir uma implementação não muito complexa, disponibilizar fácil acesso de qualquer ponto da rede, e

dispor de uma arquitetura centralizada para que as gerações e medições possam ser inicializadas simultaneamente.

Apesar de o SCGT ser utilizado especificamente através da plataforma *UNIX*, esta escolha foi viável, devido ao fato desta implementação estar bastante evoluída, permitindo a integração de ferramentas de geração de tráfego sintético de código aberto, além de oferecer segurança para a abertura de conexões remotas a serem abertas nos eventos de geração do tráfego.

Desenvolvido para validar os conceitos de medições de desempenho em um ambiente distribuído, o sistema foi capaz de atingir os objetivos traçados. Integra e manipula de forma confiável os processos de acordo com cada ferramenta de geração de tráfego sintético, permitindo que estas possam operar simultaneamente e com a melhor precisão possível. Além disso, este sistema adicionou outras funcionalidades às ferramentas IPERF e DITG, como: o agendamento dos eventos de geração de tráfego e a realização de relatórios gráficos.

9.2. Trabalhos futuros

Como o SCGT é um projeto inicial ainda é preciso evoluir em alguns aspectos, acrescentando novas funcionalidades de forma a possibilitar uma maior integração com outros recursos. A seguir seguem algumas sugestões para que futuros trabalhos possam ser desenvolvidos através deste sistema:

- Agregar mais ferramentas com diferentes destinos de perfil de tráfego, como, por exemplo, tráfegos que requerem reserva de recursos (RSVP), possibilitando o uso de QoS nos experimentos.
- Possibilitar receber o tráfego gerado em um cliente genérico, que pudesse ser utilizado como uma referência para os resultados das ferramentas. Tanto a DITG, como a IPERF só permitem a utilização do seu cliente específico. Como alternativa é possível a utilização de outro tipo de ferramentas, que apenas fazem a leitura do tráfego da rede.
- Unificar os relatórios de cada ferramenta na coleta final dos resultados, permitindo uma melhor compreensão para estudos que envolvam engenharia de tráfego.
- Disponibilizar o SCGT através de um único “pacote” de instalação, de forma que o sistema possa ser facilmente difundido pela *Web*.

Capítulo 10

Bibliografia

9.1. Referências Bibliográficas

- [ANS08] Advanced Network and Services, Inc. Disponível em: <<http://www.advanced.org/>>. Acesso em: 12 Outubro 2007.
- [AUT08] Aula – Sistemas Distribuídos II. Disponível em: <<http://www.dainf.cefetpr.br/~tacla/SDII/Cap10-01-TempoRelogios.pdf>>. Acesso em: 20 Dezembro 2008.
- [CAID08] The CAIDA Web Site. Disponível em: <<http://www.caida.org>>. Acesso em: 20 Dezembro 2007.
- [COUL08] Coulouris,G, Dollimore, J “Sistemas Distribuídos Conceitos e Projetos” Editora artmed, 4a edição, Jul,2007.
- [CSG08] Common Solutions Group. Disponível em: <<http://www.stonesoup.org/>>. Acesso em: 23 Dezembro 2007.
- [DHTM08] DHTML Tutorial. Disponível em: <<http://www.w3schools.com/dhtml/>>. Acesso em: 20 Outubro 2007.
- [DITG08] D-ITG, Distributed Internet Traffic Generator. Disponível em: <<http://www.grid.unina.it/software/ITG>>. Acesso em: 17 Outubro 2007.
- [G.692] ITUT. Recommendation G.692. “Optical interfaces for multichannel systems with optical amplifiers”. October 1998
- [G.694.1] ITUT. Recommendation G.694.1. “Spectral grids for WDM applications: DWDM frequency grid”. July 2002.

- [G.694.2] ITUT. Recommendation G.694.2. “Spectral grids for WDM applications: CWDM frequency grid”. July 2002.
- [G707] ITUT. Recommendation G.707. “Network node interface for the synchronous digital hierarchy (SDH)”. April 1991
- [IEEE08] Institute for Electrical and Electronics Engineers. Disponível em: <<http://www.ieee.org.br/>>. Acesso em: 30 Outubro 2007.
- [IETF08] Internet Engineering Task Force. Disponível em <http://www.ietf.org/>. Acesso em: 29 Outubro 2007.
- [IPF08] IPERF. Disponível em <<http://dast.nlanr.net/Projects/IPERF/>>. Acesso em: 24 Setembro 2007.
- [IPPM08] IP Performance Metrics. Disponível em <<http://www.ietf.org/html.charters/ippm-charter.html>>. Acesso em: 20 Setembro 2007.
- [IQOM08] J. A. Suruagy Monteiro¹, Edison T. L. Melo², Nelson L. Duarte Filho³, Lisandro Z. Granville⁴, Carlos A. Malcher Bastos⁵, Luís C. S. Magalhães. “GigaIQoM – Infra-estrutura de Medições para a Rede Giga (Subprojeto 2465)”. Disponível em: <<http://indico.rnp.br/getFile.py/access?contribId=3&resId=0&materialId=3&confId=33>>. Acesso em: 13 Setembro 2007.
- [INT208] Internet2. Disponível em < <http://e2epi.internet2.edu/>>. Acesso em: 13 Janeiro 2008.
- [ITUT08] Telecommunication Standardization Sector (ITU-T). Disponível em <<http://www.itu.int/ITU-T/>>. Acesso em: 22 Setembro 2007.
- [JAVA08] JavaScript MDC. Disponível em: <<http://developer.mozilla.org/en/docs/JavaScript/>>. Acesso em: 02 Novembro 2007.
- [MART05] Martins, L. “Visão Tecnológica da Rede Experimental de Alta Velocidade”, Novembro 2005. Disponível em: <<http://www.cpqd.com.br/img/13-giga-artigoforum-visao.pdf>>. Acesso em 25 Outubro 2007.
- [MGE08] MGEN – Multi Generator. Disponível em <<http://cs.itd.nrl.navy.mil/work/mgen/>>. Acesso em: 20 Outubro 2007.
- [NBR96] NBR13596. “Tecnologia de Informação – Avaliação de produto de software – Características de qualidade e diretrizes para o seu uso”, Maio 1996.
- [NLAN08] NLANR – Measurement and Network Analysis. Disponível em <<http://watt.nlanr.net/active/intro.html>>. Acesso em 12 Setembro 2007.
- [NTP08] Projeto NTP.br. Disponível em <<http://www.ntp.br/>>. Acesso em: 12 Dezembro 2007.

- [OCT08] Octave Projects. Disponível em <<http://www.gnu.org/software/octave/help-wanted.html>>. Acesso em 02 Janeiro 2008.
- [PHP08] PHP: Hypertext Prossessing. Disponível em <<http://www.php.net/>>. Acesso em: 13 Janeiro 2008.
- [PIPE08] The Internet2 E2E piPEs Project: An Interoperable Federation of Measurement Domains for Performance. Disponível em <<http://e2epi.internet2.edu/e2epipes>>. Acesso em: 15 Janeiro 2008.
- [RFC1262] Request for Comments: 1262, Vinton G. Cerf, “Guidelines for Internet Measurement Activities”, October 1991.
- [RFC1305] Request for Comments: 1305, Mills, D. L., “Network Time Protocol (Version3) Specification, Implementation and Analysis”, March 1992.
- [RFC2030] Mills, D. L., “Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI”, October 1996.
- [RFC2330] Request for Comments: 2330, J. Mahdavi, V. Paxson, G. Almes e M. Mathis. “Framework for IP Performance Metrics”, May 1998.
- [RFC2678] Request for Comments: 2678, J. Mahdavi e V. Paxson, “IPPM Metrics for Measuring Connectivity”, September 1999.
- [RFC2679] Request for Comments: 2679, G. Almes, S. Kalidindi e M. Zekauskas, “A One-way Delay Metric”, September 1999.
- [RFC2680] Request for Comments: 2680, G. Almes, S. Kalidindi e M. Zekauskas, “A One-way Packet Loss Metric for IPPM”, September 1999.
- [RFC2681] Request for Comments: 2681, G. Almes, S. Kalidindi e M. Zekauskas, “A Round-trip Delay Metric for IPPM”, September 1999.
- [RFC3393] Request for Comments: 3393, C. Demichelis e P. Chimento, “IP Packet Delay Variation Metric”, November 2002.
- [RFC791] Request For Comments 791.”Internet Protocol – Protocol Specification” DARPA Internet Program. September 1981.
- [RSBG08] Barbosa, Rodrigo. “Calculando Métricas Unidirecionais na Internet”. Trabalho de Graduação - Universidade Federal, Pernambuco, 2005. Disponível em: <<http://www.cin.ufpe.br/~tg/2004-2/rsbgb.pdf>> Acesso em: 02 Dezembro 2007.
- [RUD08] Rude and Crude. Disponível em: <<http://rude.sourceforge.net/>>. Acesso em: 02 Novembro 2007.
- [SKT08] Skitter. Disponível em: <<http://www.caida.org/tools/measurement/skitter/>>. Acesso em: 02 Dezembro 2007.

- [SSH08] Daniel J. Barrett, Ph. D., Richard E. Silverman, Robert G. Byrnes .“SSH: The Secure Shell (The Definitive Guide)”, O’Reilly, 2a Edição, May 2005.
- [SUR08] Surveyor Project. Disponível em: <<http://www.advanced.org/surveyor/>>. Acesso em: 02 Dezembro 2007.
- [TCP81] Request For Comments: 793, J. Postel, “Transmission Control Protocol”, September 1981.
- [TMG08] LAND – Laboratory for modeling, analisys and development of network and computing system. Disponível em: <araruama.land.ufjf.br>. Acesso em: 02 Outubro 2007.
- [TTM08] Test Traffic Measurements Service. Disponível em: <<http://www.ripe.net/test-traffic/>>. Acesso em: 02 Outubro 2007.
- [WIKI08] Wikipedia, a enciclopédia livre – World Wide Web. Disponível em: <<http://pt.wikipedia.org/wiki/web/>>. Acesso em: 27 Janeiro 2008.
- [WPER08] Murta, Cristina D. “Caracterização da Rede de Sincronização na Internet”. Dissertação de Mestrado - Universidade Federal do Paraná, Curitiba, 2007. Disponível em <<http://dspace.c3sl.ufpr.br/dspace/bitstream/1884/11294/1/dissertacao-pedro07survey.pdf/>>. Acesso em: 02 Fevereiro 2008.
- [XHTM08] XHTML2 Working Group Home Page. Disponível em: <<http://www.w3.org/MarkUp/>>. Acesso em 20 Janeiro 2008.

Apêndice A

Visão Tecnológica da rede GIGA

A rede GIGA é uma rede experimental que faz parte de um projeto, e tem por objetivo fornecer um ambiente capaz de atender diversos subprojetos relacionados à pesquisa e desenvolvimento em telecomunicações. Este subprojetos contam com a coordenação do RNP e do CPqD [Mart05].

Esta rede é constituída de fibras ópticas que em quase sua totalidade foram cedidas por algumas operadoras de telefonia. A estas fibras ópticas foram acoplados equipamentos que operam com tecnologia CWDM (Coarse Wavelength Division Multiplexing) [G.694.2], e DWDM (Dense Wavelength Division Multiplexing) [G.694.1], bem como de diversos elementos como transponders, multiplexadores, dentre outros.

Desde o momento de sua inauguração até hoje, a rede GIGA teve grandes avanços em números de fibras ópticas instaladas, além dos vários subprojetos que a compõem. Destes subprojetos, muitos estão em fase de conclusão tendo seus resultados validados.

Todas as configurações físicas feitas na rede GIGA, são realizadas para atender às demandas de subprojetos de pesquisa. Cada subprojeto possui suas próprias características e exigem configurações específicas de acordo com cada demanda.

Topologia da rede GIGA

Aliada as fibras ópticas se encontra a rede IP, formada de equipamentos que possuem placas MPLS, *Fast ethernet* e *Gigabit Ethernet*. Através destes equipamentos são realizados os acessos aos laboratórios nas instituições participantes do projeto GIGA, e são feitas diversas configurações dependendo do cenário de cada laboratório e de seus experimentos. Para dar suporte à rede existem ainda vários servidores para gerencia e serviços da rede.

A ligação entre o equipamento IP e o equipamento óptico é feita da seguinte forma: Cada porta GIGA do equipamento IP tem um GBIC cuja saída (Tx) encaminha os dados em um comprimento de onda de $1,31\mu\text{m}$. A saída de cada uma das GBICs de cada porta do equipamento IP é conectada a um dos *transponders* que permite a adequação da frequência do sinal de entrada para um sinal de saída compatível com o plano de frequências padronizado pelas normas G.694.1 [G.694.1] e G.694.2 [G.694.2] do ITU-T. Os *transponders* quando interligados com os equipamentos DWDM e CWDM, têm a capacidade de otimizar o uso de redes de fibra óptica, fornecendo uma infra-estrutura de meios ópticos que permite a inserção de mais de um sistema de telecomunicações, seja ele para redes de dados e/ou voz, em uma única fibra óptica.

Todos os transponders se conectam a um multiplexador que por sua vez se conecta a um amplificador. O sinal que sai deste, passa por um SOM (*Supervisor Optical Multiplexer*), que insere uma frequência de gerência (1510nm), e sai para as fibras ópticas. Para receber um sinal, um SOD (*Supervisor Optical Demultiplexer*) na estação seguinte recupera a frequência de gerência, passando-a para a recepção do módulo de gerência do equipamento, e as frequências relativas aos dados das portas vindos do equipamento IP vão para um amplificador.

Caso a estação for apenas destinada à amplificação, as frequências amplificadas se juntarão novamente à frequência de gerencia na estação em um SOM, e serão encaminhadas até a próxima estação. Se a estação tiver um equipamento que irá consumir alguns comprimentos de onda, após os sinais passarem pelo amplificador, estas vão para um demultiplexador, cujas saídas seguirão para uma entrada (Rx) de uma GBIC.

Com base nesta explicação, pode-se concluir que a topologia IP fica bastante simplificada, pois a existência de equipamentos ópticos não é percebida, não enxergando esta camada, já que é feito um enlace ponto-a-ponto (*point-to-point*) como se ambos os equipamentos IP estivessem diretamente conectados. Quando na topologia vê-se um link entre dois equipamentos, sabe-se que, ou ambos estão ligados por um cordão óptico, ou estão ligados através de um ou mais equipamentos DWDM ou CWDM.

Uma particularidade da rede GIGA, é a não existência de redundância de fibras entre Campinas e Rio de Janeiro, e internamente ao Rio de Janeiro, havendo um *loop* entre os

equipamentos IP de núcleo. Logicamente este loop está protegido por um recurso de switch para evitar *broadcast storms* na rede. No entanto, tal redundância de caminhos só existe em nível lógico (camada 2 e 3), e não na física. Isso implica que quando há rompimento de fibra ou algum eventual problema nos elementos ópticos da rede GIGA, em algum ponto entre Campinas e São Paulo ou São Paulo e Rio de Janeiro, vários enlaces caem simultaneamente, devido ao enlace ser criado pelos comprimentos de onda do DWDM ou CWDM.

Equipamentos da Rede IP

A rede IP é constituída de switches de camada 3 (*Routers Switches*), sendo estes de três modelos: *BlackDiamond 10808* (para o núcleo da rede), *BlackDiamond 6808* (para o núcleo e distribuição) e *Summit 200-24* (para o acesso).

Os módulos adquiridos para os equipamentos da rede GIGA proporcionam até 60 portas GIGA nos BD 10k, 16 ou 8 portas e mais 48 portas 10/100 nos BD 6808, e 24 portas 10/100 com mais 2 portas GIGA nos *Switch Summits*.

Topologia dos Servidores

A rede GIGA conta com vários servidores configurados para vários serviços de rede, sendo que para a sincronização dos relógios, são utilizados dois servidores em específico: o de NTP (*Network Time Protocol*) (*stratum 1 e 2*) e o DNS (*Domain Name Service*) que é utilizado para mapear endereços IP para nomes e vice-versa.

Configurações

A rede GIGA é configurada de acordo com as demandas dos grupos de pesquisas das instituições. Existem projetos que basicamente utilizam a rede GIGA para transportes de dados, voz ou imagem, existem aqueles que precisam capturar algum tipo de dado dos equipamentos, principalmente de gerência; outros precisam de acesso aos equipamentos IP para realizar algum tipo de configuração, e ainda projetos que precisam interromper certos enlaces para teste de equipamentos ópticos.

Para os subprojetos que demandam interrupção de enlace, é feito um agendamento prévio, e todos os grupos que utilizam a rede GIGA são devidamente avisados para que não ocorra nenhum problema com outros projetos. Para estes projetos, caso haja alguma necessidade de

configuração dos equipamentos IP, é estudada a demanda e realizada a configuração. Para os subprojetos que demandam capturar dados na rede GIGA, é criada uma VLAN específica para estes grupos. Para os subprojetos que usam a rede GIGA para transporte em alta velocidade, são feitas VLAN's de interconexão entre os laboratórios das instituições parceiras. Internamente ao laboratório algumas vezes é criada uma VLAN e um roteamento estático.

VLAN's

Atualmente a maioria dos serviços oferecidos para os subprojetos que necessitam de conectividade à rede GIGA, são provisionados através de VLAN's. Existem vários tipos de VLAN's criadas na rede GIGA:

- VLAN de Enlace: São as VLAN's de interconexão entre dois switches, ponto-a-ponto, criando um enlace entre os dois equipamentos.
- VLAN de Gerência: São VLAN's que possuem máquinas de gerência da rede GIGA, tanto IP quanto óptica.
- VLAN de Subprojeto: É qualquer VLAN criada para atender algum subprojeto.
- VLAN Temporária: São VLAN's feitas para algum evento ou teste temporário.

Apêndice B

Código Fonte do Módulo CGT

Ferramenta IPERF

<?

```

//Simplificação do diretório
$local[$servidor] = "/usr/src/IPERF-2.0.2/src/";
//Simplificação da identificação do arquivo
$arquivo = "IPERF_" .date("Y-m-d_H-i"). ".txt";

```

Perfil TCP – Modo Agendado

```

if(isset($_POST["submitTCPag"])) { //TCP
//Script para servidor e cliente
$arquiv = "agenda_" .date("Y-m-d_H-i"). ".txt";
$linhaSq1 = 'ssh gerador@'. $_POST['dest'] .' screen -d -m '. $local[$servidor] .'IPERF -s ' .
"\n";
$linhaCq1 = 'ssh gerador@'. $_POST['org'] .' IPERF -c'. $_POST['dest'];
$linhaCq1 .= '-w' . $_POST["janela"];

```

```

$linhaCq1 .= '-M' . $_POST["mtu"];
    if($_POST["geracao"]==1) {
        $linhaCq1 .= '-i1';
    } else {
        $linhaCq1 .= '-i' . $_POST["periodicidade"];    }
    echo $linhaCq1 .= '-t' . $_POST["duracao"]*60;
    if($_POST["paralelo"] > 1) {
        $linhaCq1 .= '-P' . $_POST["paralelo"];
        $linhaSq1 .= '-P' . $_POST["paralelo"];    }
$linhaCq1 .= " >/home/gerador/basemichele/ag1_$arquiv". "\n";
    //Script para Desabilitar processos na origem e destino
$linhaFq = 'ssh gerador@'. $_POST["dest"] .' killall IPERF'. "\n";
$linhaFq3 = 'ssh gerador@'. $_POST["org"] .' killall IPERF'. "\n";
    //Script para ver hora local
$arquivod = date ("H:i:s"). "";
echo "</p></pre><p><center>HORA LOCAL deste SERVIDOR: $arquivod</p>";
    //Recebe variável de hora do agendamento
$linhaq = $_POST["min"] . ' ';
$linhax = $_POST["min"];
$linhaq .= $_POST["hora"] . ' * * * *';
$linhaqw = $_POST["min"]+$_POST["duracao"];
    //Contagem para inicializar o servidor um minuto antes
$linhaqw1 = ($_POST["min"])-1;
    //Contagem para Terminar os processos, copiar e decodificar no somatório dos tempos
$linhaqw2 = (($_POST["min"]+1)+$_POST["duracao"]);
    //Se o minuto for menor que 10, recebe número 0 na frente
if ($linhaqw1 < '10') {
$linhaqw1 ='0';
$linhaqw1 = ($_POST["min"])-1;
    $linhaqw1 .= ' ';
    $linhaqw1 .= $_POST["hora"] . ' * * * *'; }
else{
$linhaqw1 = ($_POST["min"])-1;
$linhaqw1 .= ' ';
$linhaqw1 .= $_POST["hora"] . ' * * * *';}

```

```

if ($linhaqw2 < '10') {
$linhaqw2 ='0';
echo $linhaqw2 . = (($_POST["min"])+($_POST["duracao"])+1);
$linhaqw2 .= ' ';
$linhaqw2 .= $_POST["hora"] . ' * * * *'; }
else{
$linhaqw2 = (($_POST["min"])+($_POST["duracao"])+1);
$linhaqw2 .= ' ';
$linhaqw2 .= $_POST["hora"] . ' * * * *';
$linhaqw2 ='0';}
if ($linhaqq < '10') {
$linhaqq = ($_POST["min"]+$_POST["duracao"]);
$linhaqq .= ' ';
$linhaqq .= $_POST["hora"] . ' * * * *';}
else{
$linhaqq .= ($_POST["min"]+$_POST["duracao"]);
$linhaqq .= ' ';
$linhaqq .= $_POST["hora"] . ' * * * *';}
$linharel = $_POST["hora"]. ' : ';
$linharel . = (($_POST["min"])+($_POST["duracao"])+1);
//Apresenta nome do primeiro relatório
echo "</p></pre><p>Relatório do primeiro teste agendado : ag1_$arquiv<br />";
echo "</p></pre><p>Acessar este relatório através do link RELATORIO após as: $linharel <br
/></p></pre><p>";
// Script para reinicializar a rotina do cron
$linhacron = 'ssh gerador@10.24.65.7 crond restart';
//Abre o arquivo colocando string apartir da ultima linha do arquivo apache
$file= "/var/spool/cron/crontabs/apache";
$fileopen = fopen($file, "a+");
fwrite($fileopen,$linhaqw1 . $linhaSq1 . $linhaq . $linhaCq1 . $linhaqw2 . $linhaFq . $linhaqw2
. $linhaFq3);
fclose($fileopen);
//Se hover, inicializa segundo fluxo com os mesmos passos do processo anterior
if($_POST["dest1"] > '0') {

```

```

$linhaSq = 'ssh gerador@'. $_POST['dest1'] .' screen -d -m '. $local[$servidor] .'IPERF -s ' .
"\n";
$linhaCq = 'ssh gerador@'. $_POST['org1'] .' IPERF -c'.'$_POST['dest1'];
$linhaCq .= ' -w' . $_POST["janela"];
$linhaCq .= ' -M' . $_POST["mtu"];
    if($_POST["geracao"]==1) {
        $linhaCq .= ' -i1';
    } else {
        $linhaCq .= ' -i' . $_POST["periodicidade"];}
        $linhaCq .= ' -t' . $_POST["duracao"]*60;

    if($_POST["paralelo"] > 1) {
        $linhaCq .= ' -P' . $_POST["paralelo"];
        $linhaSq .= ' -P' . $_POST["paralelo"];}
    $linhaCq .= " >/home/gerador/basemichele/ag2_$arquiv" . "\n";
echo "</p></pre><p>Relatório do segundo teste agendado : ag2_$arquiv<br />";
    //Horário marcado para buscar o relatório
echo "</p></pre><p>Acessar este relatório através do link RELATORIO após as: $linharel <br
/></p></pre><p>";
echo $linhar1 = $_POST["hora"] . ':' . $_POST["min"] ;
$file= "/var/spool/cron/crontabs/apache";
$fileopen = fopen($file, "a+");
fwrite($fileopen,$linhaqw1 . $linhaSq . $linhaq . $linhaCq );
fclose($fileopen);}

    //Restart linha de cron do SCGT executando o arquivo
system ($linhacron);}]

```

Perfil TCP – Modo Instantâneo

```

if(isset($_POST["submitTCP"])) { //TCP
    //Script para inicializa servidor e cliente
    $arquiv = "TCP_" .date("Y-m-d_H-i"). ".txt";
    $linhaSq1 = 'ssh gerador@'. $_POST['dest'] .' screen -d -m '. $local[$servidor] .'IPERF -s ';
    $linhaCq1 = 'IPERF -c'.'$_POST['dest'];

```

```

$linhaCq1 .= '-w' . $_POST["janela"];
$linhaCq1 .= '-M' . $_POST["mtu"];
if($_POST["geracao"]==1) {
    $linhaCq1 .= '-i1';
} else {
    $linhaCq1 .= '-i' . $_POST["periodicidade"];    }
$linhaCq1 .= '-t' . $_POST["duracao"]*60;
if($_POST["paralelo"] > 1) {
    $linhaCq1 .= '-P' . $_POST["paralelo"];
    $linhaSq1 .= '-P' . $_POST["paralelo"];    }
$linhaSq1 .= ';';
$linhaCq1 .= ">/home/gerador/basemichele/ag1_$arquiv" . "\n";
    //Script para desabilitar processos
$linhaFq = 'ssh gerador@'. $_POST["dest"] .' killall IPERF;';
$linhaFq3 = ' killall IPERF' . "\n";
$linhaFq1 = 'ssh gerador@'. $_POST["dest1"] .' killall IPERF;';
$linhaFq31 = ' killall IPERF' . "\n";
    //Recebe variável de hora local
$linhaH = date ("H");
$linhaM = date ("i");
$linhaSeg = date ("s");
$linhaHq = date ("H");
    //Faz contagem de tempo para o processo de inicialização
$linhaMi = ($linhaM)+1;
$linhaHi = (((($linhaM)+1)+$_POST["duracao"]);
$linhaHq1 = (((($linhaM)+1)+$_POST["duracao"]);
$linhaqw3 = ($linhaH).' :';
$linhaqw3 .= ($linhaM)+1;
    //Se o minuto for menor que 10, recebe número 0 na frente
if (($linhaMi) < '10') {
    $linhaHq = '0';
    $linhaHq .= ($linhaMi);
    $linhaHq .= ' ';
    $linhaHq .= ($linhaH) . ' * * * * '; }
elseif (($linhaMi) > '59') {

```

```

$linhaHq ='00';
$linhaHq .= ' ';
$linhaHq .=( $linhaH)+1;
$linhaHq .= ' * * * '; }
else{
$linhaHq = ($linhaMi);
$linhaHq .= ' ';
$linhaHq .=( $linhaH) . ' * * * ';}
//Termina processo no cliente
if (($linhaHq1) < '10') {
$linhaHq1 ='0';
$linhaHq1 .=( $linhaHi);
$linhaHq1 .= ' ';
$linhaHq1 .=( $linhaH);
$linhaHq1 .= ' * * * ';}
elseif (($linhaHq1) > '59') {
$linhaHq1 = '0';
$linhaHq1 .=( $linhaHi)-60;
$linhaHq1 .= ' ';
$linhaHq1 .=( $linhaH)+1;
$linhaHq1 .= ' * * * '; }
else{
$linhaHq1 = ($linhaHi);
$linhaHq1 .= ' ';
$linhaHq1 .=( $linhaH) . ' * * * ';}
//Script para copiar relatorio do cliente
$linhaSend1 = 'scp gerador@'. $_POST["org"] .":/home/gerador/basemichele/ag1_$arquiv
/home/gerador/basemichele/";
$linhaSend11 = 'scp gerador@'. $_POST["org1"] .":/home/gerador/basemichele/ag11_$arquiv
/home/gerador/basemichele/";
$linhaSendX = 'scp gerador@'. $_POST["org"] .":/home/gerador/basemichele/ntp_$arquiv
/home/gerador/basemichele/";
$linhaSendX1 = 'scp gerador@'. $_POST["org1"] .":/home/gerador/basemichele/ntp1_$arquiv
/home/gerador/basemichele/";

```

```

$linhaX      =          "date          +%H:%M:%S:%N--nanosegundos          >
/home/gerador/basemichele/ntp_$arquiv"."n";
$linhaX1     =          "date          +%H:%M:%S:%N--nanosegundos          >
/home/gerador/basemichele/ntp1_$arquiv"."n";
//Script para remover o arquivo de cron
$linhaRem = 'ssh gerador@'. $_POST['org']. ' rm /var/spool/cron/crontabs/IPERF;';
$linhaRem1 = 'ssh gerador@'. $_POST['org1']. ' rm /var/spool/cron/crontabs/IPERF;';
//Script para copiar arquivo de cron para a máquina geradora
$Send      = 'scp /home/gerador/basemichele/IPERF ssh gerador@'. $_POST["org"]
.:/var/spool/cron/crontabs;';
$Send1     = 'scp /home/gerador/basemichele/im/IPERF ssh gerador@'. $_POST["org1"]
.:/var/spool/cron/crontabs;';
//Script que Reinicia Cron na máquina geradora.
$linhaCron = 'ssh gerador@'. $_POST['org']. ' crond restart;';
$linhaCron1 = 'ssh gerador@'. $_POST['org1']. ' crond restart;';
//Faz Somatorio para espera de tempo
$Div= 60-($linhaSeg);
$tempo=( $_POST["duracao"])*60+($f);
$f=($Div)+($tempo)+1;
//Script para segund fluxo
$linhaSq11 = 'ssh gerador@'. $_POST['dest1'] .' screen -d -m '. $local[$servidor] .'IPERF -s ';
$linhaCq11 = 'IPERF -c'. $_POST['dest1'];
$linhaCq11 .= ' -w' . $_POST["janela"];
$linhaCq11 .= ' -M' . $_POST["mtu"];
if($_POST["geracao"]==1) {
    $linhaCq11 .= ' -i1';
} else {
$linhaCq11 .= ' -i' . $_POST["periodicidade"];}
$linhaCq11 .= ' -t' . $_POST["duracao"]*60;
if($_POST["paralelo"] > 1) {
    $linhaCq11 .= ' -P' . $_POST["paralelo"];
    $linhaSq11 .= ' -P' . $_POST["paralelo"];}
$linhaSq11 .= ';';
$linhaCq11 .= ">/home/gerador/basemichele/ag11_$arquiv"."n";
//Faz linha de cron

```

```

$file="/home/gerador/basemichele/IPERF";
$fileopen = fopen($file,"r+");
fwrite($fileopen,$linhaHq . $linhaCq1 . $linhaHq1 . $linhaFq3 . $linhaHq . $linhaX);
    //Faz linha de cron para o segundo fluxo
if($_POST["dest1"] > '0') {
$file1="/home/gerador/basemichele/im/IPERF";
$fileopen1 = fopen($file1,"r+");
fwrite($fileopen1,$linhaHq . $linhaCq11 . $linhaHq1 . $linhaFq31 . $linhaHq . $linhaX1);}
    //Executa Servidor e arquivo de cron
system ($linhaSq1 . $linhaSq11 . $Send . $Send1 . $linhaCron . $linhaCron1);
    //Espera um tempo
$linhaAh ='sleep '. $f.';';
    //Hora local deste servidor
$arquivod = date ("H:i:s"). "";
    //Apresenta hora de inicialização do servidor
echo "</p></pre><p><center>Inicialização do(s) Servidor(es) às: $arquivod</p>";
    //Executa Processos de pegar arquivos
system ($linhaAh . $linhaFq1 . $linhaFq . $linhaSend11 . $linhaRem1 . $linhaSend1 . $linhaRem
. $linhaSendX . $linhaSendX1);
if($_POST["dest1"] > '0') {
    //Apresenta em tela
echo "<p> Resultado do fluxo 2:<br><pre>";
    echo file_get_contents("/home/gerador/basemichele/ag11_". $arquiv);
    unlink("/home/gerador/basemichele/tcp12_". $arquivo);
echo "</p></pre><p>Relatório do segundo fluxo : ag11_ $arquiv<br />";
echo "</p></pre><p> Hora de envio do segundo fluxo pela máquina de destino:<br><pre>";
    echo file_get_contents("/home/gerador/basemichele/ntp1_". $arquiv);}
echo "</p></pre><p>_____";
echo "</p></pre><p> Hora de envio do primeiro fluxo pela máquina de destino:<br><pre>";
    echo file_get_contents("/home/gerador/basemichele/ntp_". $arquiv);
    //Apresenta em tela
echo "</p></pre><p> Resultado do fluxo 1:<br><pre>";
    echo file_get_contents("/home/gerador/basemichele/ag1_". $arquiv);
    unlink("/home/gerador/basemichele/tcp1_". $arquivo);
echo "</p></pre><p>_____";

```

```
echo "</p></pre><p>Relatório do primeiro fluxo : ag1_$arquiv<br />"; }
```

Perfil UDP – Modo Agendado

```
if(isset($_POST["submitUDPag"])) { //UDP
$arquiv = "agenda_" . date("Y-m-d_H-i"). ".txt";
    //Cria Script para cliente e servidor
$linhaSq = 'ssh gerador@'. $_POST['dest2'] .' screen -d -m '. $local[$servidor] .'IPERF -u -s';
$linhaCq = 'ssh gerador@'. $_POST['org2'] .' IPERF -u -c'$_POST['dest2'];
if($_POST["geracaoUDP"]==1) {
    $linhaCq .= ' -i1';
} else {
    $linhaCq .= ' -i' . $_POST["periodicidadeUDP"];
    $linhaCq .= ' -t' . $_POST["duracaoUDP"]*60;
if($_POST["direcao"]==2) {
    $linhaCq .= ' -d' . $_POST["qtdconexoes"];
    $linhaCq .= ' -p' . $_POST["porta"];
    $linhaSq .= ' -p' . $_POST["porta"]. "\n";
if($_POST["paralelo"] > 1) {
    $linhaCq .= ' -P' . $_POST["paralelo"];
    $linhaSq .= ' -P' . $_POST["paralelo"];
    $linhaCq .= ' -f' . $_POST["unidade"];
    $linhaCq .= ' -b' . $_POST["banda"];
    $linhaCq .= ' -l' . $_POST["pacoteUDP"];
$linhaCq .= " >/home/gerador/basemichele/ag_$arquiv". "\n";
    //Script para reinicializar rotina de cron
$linhacron = 'ssh gerador@10.24.65.7 crond restart';
    //Script para finalizar processos
$linhaFh = 'ssh gerador@'. $_POST['dest2'] .' killall IPERF' . "\n";
$linhaFh3 = 'ssh gerador@'. $_POST['org2'] .' killall IPERF' . "\n";
$arquivod = date ("H:i:s"). "";
echo "</p></pre><p><center>HORA LOCAL deste SERVIDOR: $arquivod</p>";
    //Recebe variável de hora do agendamento
$linhaq = $_POST["min"] . ' ';
$linhax = $_POST["min"];
$linhaq .= $_POST["hora"] . ' * * * * ';
```

```

$linhaqw = $_POST["min"]+$_POST["duracaoUDP"];
    //Inicializa o servidor um minuto antes
$linhaqw1 = ($_POST["min"])-1;
    //Termina os processos, copia e decodifica no somatório dos tempos
$linhaqw2 = (($_POST["min"]+1)+$_POST["duracaoUDP"]);
    //Se o minuto for menor que 10, recebe número 0 na frente
if ($linhaqw1 < '10') {
$linhaqw1 ='0';
$linhaqw1 . = ($_POST["min"])-1;
$linhaqw1 . = ' ';
$linhaqw1 . = $_POST["hora"] . ' * * * * '; }
else{
$linhaqw1 = ($_POST["min"])-1;
$linhaqw1 . = ' ';
$linhaqw1 . = $_POST["hora"] . ' * * * * ';}
if ($linhaqw2 < '10') {
$linhaqw2 ='0';
$linhaqw2 . = (($_POST["min"])+($_POST["duracaoUDP"])+1);
$linhaqw2 . = ' ';
$linhaqw2 . = $_POST["hora"] . ' * * * * '; }
else{
$linhaqw2 = (($_POST["min"])+($_POST["duracaoUDP"])+1);
$linhaqw2 . = ' ';
$linhaqw2 . = $_POST["hora"] . ' * * * * ';
$linhaqq ='0';}
if ($linhaqq < '10') {
$linhaqq = ($_POST["min"]+$_POST["duracaoUDP"]);
$linhaqq . = ' ';
$linhaqq . = $_POST["hora"] . ' * * * * ';}
else{
$linhaqq . = ($_POST["min"]+$_POST["duracaoUDP"]);
$linhaqq . = ' ';
$linhaqq . = $_POST["hora"] . ' * * * * ';}
$linharel = $_POST["hora"]. ' : ';
$linharel . = (($_POST["min"])+($_POST["duracaoUDP"])+1);

```

```

//Apresenta nome do primeiro relatório
echo "</p></pre><p>Arquivo do primeiro relatório agendado : ag_<math>\$</math>arquiv<br />";
echo "</p></pre><p>Acessar este relatório através do link RELATORIO após as: <math>\$</math>linharel <br /></p></pre><p>";

//Coloca arquivo para apresentação no link Arquivo
$fname = "/home/gerador/basemichele/ag_". $arquiv;

///Restarta a rotina do cron
$linhacron = 'ssh gerador@10.24.65.7 crond restart';

//Abre o arquivo colocando string apartir da ultima linha do arquivo apache
$file= "/var/spool/cron/crontabs/apache";
$fileopen = fopen($file, "a+");
fwrite($fileopen,$linhaqw1 . $linhaSq . $linhaq . $linhaCq . $linhaqw2 . $linhaFh . $linhaqw2 .
$linhaFh3);
fclose($fileopen);

//Inicializa 2 processos simultaneos com os mesmos passos do processo anterior
if($_POST["dest3"] > '0') {
$linhaSq1 = 'ssh gerador@'. $_POST['dest3'] . ' screen -d -m ' . $local[$servidor] . 'IPERF -u -s';
$linhaCq1 = 'ssh gerador@'. $_POST['org3'] . ' IPERF -u -c'. $_POST['dest3'];
if($_POST["geracaoUDP"]==1) {
    $linhaCq1 .= ' -i1';
} else {
    $linhaCq1 .= ' -i' . $_POST["periodicidadeUDP"];
    $linhaCq1 .= ' -t' . $_POST["duracaoUDP"]*60;
if($_POST["direcao"]==2) {
    $linhaCq1 .= ' -d' . $_POST["qtdconexoes"];
    $linhaCq1 .= ' -p' . $_POST["porta"];
    $linhaSq1 .= ' -p' . $_POST["porta"]. "\n";
if($_POST["paralelo"] > 1) {
    $linhaCq1 .= ' -P' . $_POST["paralelo"];
    $linhaSq1 .= ' -P' . $_POST["paralelo"];
    $linhaCq1 .= ' -f' . $_POST["unidade"];
    $linhaCq1 .= ' -b' . $_POST["banda"];
    $linhaCq1 .= ' -l' . $_POST["pacoteUDP"];
$linhaCq1 .= " >/home/gerador/basemichele/ag1_<math>\$</math>arquiv" . "\n";
$linhaFh1 = 'ssh gerador@'. $_POST['dest3'] . ' killall IPERF' . "\n";

```

```

$linhaFh31 = 'ssh gerador@'. $_POST['org3'] . ' killall IPERF' . "\n";
echo "</p></pre><p>Arquivo do segundo relatório agendado : ag1_ $arquiv<br />";
echo "</p></pre><p>Acessar este relatório através do link RELATORIO após as: $linharel <br
/></p></pre><p>";
    //Coloca arquivo para apresentação no link Arquivo
$fname = "/home/gerador/basemichele/ag1_". $arquiv;
    //Restarta a rotina do cron
$linhacron = 'ssh gerador@10.24.65.7 crond restart';
    //Abre o arquivo colocando string apartir da ultima linha do arquivo apache
$file= "/var/spool/cron/crontabs/apache";
$fileopen = fopen($file, "a+");
fwrite($fileopen,$linhaqw1 . $linhaSq1 . $linhaq . $linhaCq1 . $linhaqw2 . $linhaFh1 .
$linhaqw2 . $linhaFh31);
fclose($fileopen);}
system ($linhacron);}

```

Perfil UDP – Modo Instantâneo

```

elseif(isset($_POST["submitUDP"])) {
    //Script para I]inicializa servidor e cliente
$arquiv = "UDP_" .date("Y-m-d_H-i"). ".txt";
$linhaSq1 = 'ssh gerador@'. $_POST['dest2'] . ' screen -d -m '. $local[$servidor] .'IPERF -u -s ';
$linhaCq1 = 'IPERF -u -c'. $_POST['dest2'];
if($_POST["geracaoUDP"]==1) {
    $linhaCq1 .= ' -i1';
} else {
    $linhaCq1 .= ' -i' . $_POST["periodicidadeUDP"];
    $linhaCq1 .= ' -t' . $_POST["duracaoUDP"]*60;
if($_POST["direcao"]==2) {
    $linhaCq1 .= ' -d' . $_POST["qtdconexoes"];
    $linhaCq1 .= ' -p' . $_POST["porta"];
    $linhaSq1 .= ' -p' . $_POST["porta"];
if($_POST["paralelo"] > 1) {
    $linhaCq1 .= ' -P' . $_POST["paralelo"];

```

```

$linhaSq1 .= '-P' . $_POST["paralelo"]; }
$linhaCq1 .= '-f' . $_POST["unidade"];
$linhaCq1 .= '-b' . $_POST["banda"];
$linhaCq1 .= '-l' . $_POST["pacoteUDP"];

$linhaSq1 .= ';';
$linhaCq1 .= " >/home/gerador/basemichele/ag1_$arquiv" . "\n";
    //Script para desabilitar processos
$linhaFq = 'ssh gerador@' . $_POST["dest2"] . ' killall IPERF';
$linhaFq3 = ' killall IPERF' . "\n";
$linhaFq1 = 'ssh gerador@' . $_POST["dest3"] . ' killall IPERF';
$linhaFq31 = ' killall IPERF' . "\n";
    //Recebe variável de hora local
$linhaH = date ("H");
$linhaM = date ("i");
$linhaSeg = date ("s");
$linhaHq = date ("H");
    //Contagem de tempo para inicializar processo no cliente
$linhaMi = ($linhaM)+1;
$linhaHi = (((($linhaM)+1)+$_POST["duracaoUDP"]);
$linhaHq1 = (((($linhaM)+1)+$_POST["duracaoUDP"]);
$linhaqw3 = ($linhaH) . ':';
$linhaqw3 .= ($linhaM)+1;
    //Se o minuto for menor que 10, recebe número 0 na frente
if (($linhaMi) < '10') {
$linhaHq = '0';
$linhaHq .= ($linhaMi);
$linhaHq .= ' ';
$linhaHq .= ($linhaH) . ' * * * '; }
elseif (($linhaMi) > '59') {
$linhaHq = '00';
$linhaHq .= ' ';
$linhaHq .= ($linhaH)+1;
echo $linhaHq .= ' * * * '; }
else {
$linhaHq = ($linhaMi);

```

```

$linhaHq .= ' ';
$linhaHq .= ($linhaH) . ' * * * *';}
    //Termina processo no cliente
if (($linhaHq1) < '10') {
$linhaHq1 = '0';
$linhaHq1 .= ($linhaHi);
$linhaHq1 .= ' ';
$linhaHq1 .= ($linhaH);
$linhaHq1 .= ' * * * *';}
elseif (($linhaHq1) > '59') {
$linhaHq1 = '0';
$linhaHq1 .= ($linhaHi)-60;
$linhaHq1 .= ' ';
$linhaHq1 .= ($linhaH)+1;
echo $linhaHq1 .= ' * * * *'; }
else{
$linhaHq1 = ($linhaHi);
$linhaHq1 .= ' ';
$linhaHq1 .= ($linhaH) . ' * * * *';}
    //Script para copiar relatorio do cliente
$linhaSend1 = 'scp gerador@'. $_POST["org2"] .":/home/gerador/basemichele/ag1_$arquiv
/home/gerador/basemichele/;";
$linhaSend11 = 'scp gerador@'. $_POST["org3"] .":/home/gerador/basemichele/ag11_$arquiv
/home/gerador/basemichele/;";
    //Script para copiar arquivo de hora das máquinas geradoras
$linhaSendX = 'scp gerador@'. $_POST["org2"] .":/home/gerador/basemichele/ntp_$arquiv
/home/gerador/basemichele/;";
$linhaSendX1 = 'scp gerador@'. $_POST["org3"] .":/home/gerador/basemichele/ntp1_$arquiv
/home/gerador/basemichele/;";
$linhaX      =          "date          +%H:%M:%S:%N--nanosegundos          >
/home/gerador/basemichele/ntp_$arquiv"."\\n";
$linhaX1     =          "date          +%H:%M:%S:%N--nanosegundos          >
/home/gerador/basemichele/ntp1_$arquiv"."\\n";
    //Script para remover o arquivo de cron
$linhaRem = 'ssh gerador@'. $_POST['org2']. ' rm /var/spool/cron/crontabs/gerador;';

```

```

$linhaRem1 = 'ssh gerador@'. $_POST['org3']. ' rm /var/spool/cron/crontabs/gerador;';
    //Copia linha de cron ao cliente
$Send = 'scp /home/gerador/basemichele/gerador ssh gerador@'. $_POST["org2"]
.:/var/spool/cron/crontabs;';
$Send1 = 'scp /home/gerador/basemichele/im/gerador ssh gerador@'. $_POST["org3"]
.:/var/spool/cron/crontabs;';
    //Script para reiniciar Cron no cliente
$linhaCron = 'ssh gerador@'. $_POST['org2']. ' crond restart;';
$linhaCron1 = 'ssh gerador@'. $_POST['org3']. ' crond restart;';
    //Faz Somatorio para espera de tempo
$Div= 60-($linhaSeg);
$tempo=( $_POST["duracaoUDP"])*60+($f);
$f=( $Div)+($tempo)+1;
    //Inicializa servidor e cliente do segundo fluxo com os mesmos passos anterior
$linhaSq11 = 'ssh gerador@'. $_POST['dest3'] . ' screen -d -m ' . $local[$servidor] . 'IPERF -u -s ';
$linhaCq11 = 'IPERF -u -c'. $_POST['dest3'];
if($_POST["geracaoUDP"]==1) {
    $linhaCq11 .= ' -i1';
} else {
    $linhaCq11 .= ' -i' . $_POST["periodicidadeUDP"];
    $linhaCq11 .= ' -t' . $_POST["duracaoUDP"]*60;
if($_POST["direcao"]==2) {
    $linhaCq11 .= ' -d' . $_POST["qtdconexoes"];
    $linhaCq11 .= ' -p' . $_POST["porta"];
    $linhaSq11 .= ' -p' . $_POST["porta"];
if($_POST["paralelo"] > 1) {
    $linhaCq11 .= ' -P' . $_POST["paralelo"];
    $linhaSq11 .= ' -P' . $_POST["paralelo"];}
    $linhaCq11 .= ' -f' . $_POST["unidade"];
    $linhaCq11 .= ' -b' . $_POST["banda"];
    $linhaCq11 .= ' -l' . $_POST["pacoteUDP"];
$linhaSq11 .= ';';
$linhaCq11 .= " >/home/gerador/basemichele/ag11_Sarquiv" . "\n";
    //Faz linha de cron
$file="/home/gerador/basemichele/gerador";

```

```

$fileopen = fopen($file,"r+");
fwrite($fileopen,$linhaHq . $linhaCq1 . $linhaHq1 . $linhaFq3. $linhaHq . $linhaX);
    //Faz linha de cron do segundo fluxo
if($_POST["dest3"] > '0') {
$file1="/home/gerador/basemichele/im/gerador";
$fileopen1 = fopen($file1,"r+");
fwrite($fileopen1,$linhaHq . $linhaCq11 . $linhaHq1 . $linhaFq31 . $linhaHq . $linhaX1);}
    //Hora local deste servidor
$arquivod = date ("H:i:s"). "";
echo "</p></pre><p><center>Inicialização do(s) Servidor(es) às: $arquivod</p>";
echo "</p></pre><p>_____</p></pre><p>";
    //Executa Servidor e rotina de cron
system ($linhaSq1. $linhaSq11 . $Send . $Send1 . $linhaCron . $linhaCron1);
    //Espera um tempo
$linhaAh ='sleep '.$f.';';
    //Executa Processos de pegar arquivos
system ($linhaAh . $linhaFq1 . $linhaFq. $linhaSend11 . $linhaRem1 . $linhaSend1 .
$linhaRem. $linhaSendX . $linhaSendX1);
if($_POST["dest3"] > '0') {
    //Apresenta em tela resultado do segundo fluxo
echo "<p> Resultado do fluxo 2:<br><pre>";
    echo file_get_contents("/home/gerador/basemichele/ag11_". $arquiv);
    unlink("/home/gerador/basemichele/tcp12_". $arquivo);
echo "</p></pre><p>Relatório do segundo fluxo : ag11_ $arquiv<br />";
echo "</p></pre><p> Hora de envio do segundo fluxo pela máquina de destino:<br><pre>";
echo file_get_contents("/home/gerador/basemichele/ntp1_". $arquiv);}
echo "</p></pre><p>_____";
echo "</p></pre><p> Hora de envio do primeiro fluxo pela máquina de destino:<br><pre>";
echo file_get_contents("/home/gerador/basemichele/ntp_". $arquiv);
    //Apresenta em tela relatório do primeiro fluxo
echo "</p></pre><p> Resultado do fluxo 1:<br><pre>";
echo file_get_contents("/home/gerador/basemichele/ag1_". $arquiv);
unlink("/home/gerador/basemichele/tcp1_". $arquivo);
echo "</p></pre><p>Relatório do primeiro fluxo : ag1_ $arquiv<br />"; }
?>

```

Ferramenta DITG

```
<?
//Cria arquivos com marcacao de tempo
$arquivo = "ditg_" .date("Y-m-d_H-i"). ".txt";
$arquivod = date ("H:i:s"). "";
//Simplificaco do diretorio do servidor
$local[$servidor] = "/usr/src/D-ITG-2.6.1b/bin/";
```

Perfil Personalizado – Modo Agendado

```
if(isset($_POST["submitDITGag"])) {
    //Marca o tempo do arquivo
    $arquiv = "agend_" .date("Y-m-d_H-i"). ".txt";
    // ditg tempo em mseg. Granularidade foi reduzida a seg.
    $tempo=( $_POST["duracao"]/60000);
    //Script para linhas de comando do cliente e servidor
    $linhaSq = 'ssh gerador@'. $_POST["dest"] .' screen -d -m '. $local[$servidor] . "ITGRecv -l
/home/basemichele/ag_ $arquiv" ."\n";
    $linhaCq = 'ssh gerador@'. $_POST["org"] .' screen -d -m '. $local[$servidor] .'ITGSend';
    $linhaCq .= ' -a ' . $_POST["dest"] ;
    if($_POST["metrica"] == '2') {
        $linhaCq .= ' -m RTTM';    }
    $linhaCq .= ' -t ' . $_POST["duracao"]*60000;
    $linhaCq .= ' -C ' . $_POST["pacote"];
    $linhaCq .= ' -rp ' . $_POST["portaD"];
    $linhaCq .= ' -sp ' . $_POST["portaO"];
    $linhaCq .= ' -T ' . $_POST["protocolo"];
    if($_POST["geracao"] == 'U') {
        $linhaCq .= ' -u ' . $_POST["valoresU"];
    } elseif($_POST["geracao"] == 'P') {
        $linhaCq .= ' -O ' . $_POST["pctP"];
    } else {
        $linhaCq .= ' -c ' . $_POST["tampac"] ."\n";}
    //Script para copiar arquivo
```

```

$linhaGetq = 'scp gerador@'. $_POST["dest"] .":/home/basemichele/ag_$arquiv
/home/gerador/basemichele/" ."\n";
$linhaGetq1 = "/home/gerador/basemichele/ITGDec /home/gerador/basemichele/ag_$arquiv
>ag1_$arquiv" ."\n";
$linhaE = "cp /var/www/ag1_$arquiv /home/gerador/basemichele" ."\n";
//Script para desabilitar o processo em execução no servidor
$linhaFq = 'ssh gerador@'. $_POST["dest"] .' killall ITGRecv' ."\n";
    //Recebe variável de hora do agendamento
$linhaq = $_POST["min"] . ' ';
$linhax = $_POST["min"];
$linhaq .= $_POST["hora"] . ' * * * ';
$linhaqw = $_POST["min"]+$POST["duracao"];
    //Contagem - Inicializa o servidor um minuto antes
$linhaqw1 = ($_POST["min"])-1;
    //Contagem - Termina os processos, copia e decodifica no somatório dos tempos
$linhaqw2 = (($_POST["min"]+1)+$_POST["duracao"]);
    //Se o minuto for menor que 10, recebe número 0 na frente para formato da rotina de cron
if ($linhaqw1 < '10') {
$linhaqw1 ='0';
$linhaqw1 . = ($_POST["min"])-1;
$linhaqw1 . = ' ';
$linhaqw1 . = $_POST["hora"] . ' * * * '; }
else {
$linhaqw1 = ($_POST["min"])-1;
$linhaqw1 . = ' ';
$linhaqw1 . = $_POST["hora"] . ' * * * ';}
if ($linhaqw2 < '10') {
$linhaqw2 ='0';
$linhaqw2 . = (($_POST["min"])+($_POST["duracao"])+1);
$linhaqw2 . = ' ';
$linhaqw2 . = $_POST["hora"] . ' * * * '; }
else {
$linhaqw2 = (($_POST["min"])+($_POST["duracao"])+1);
$linhaqw2 . = ' ';
$linhaqw2 . = $_POST["hora"] . ' * * * ';

```

```

$linhaqq ='0';}
if ($linhaqq < '10') {
$linhaqq = ($_POST["min"]+$_POST["duracao"]);
$linhaqq .= ' ';
$linhaqq .= $_POST["hora"] . ' * * * *';}
else{
$linhaqq .= ($_POST["min"]+$_POST["duracao"]);
$linhaqq .= ' ';
$linhaqq .= $_POST["hora"] . ' * * * *';}
$linharel = $_POST["hora"]. ': ':';
$linharel .= (($_POST["min"])+($_POST["duracao"])+1);
    //Recebe hora local deste servidor
$arquivod = date ("H:i:s"). "";
echo "</p></pre><p><center>HORA LOCAL deste SERVIDOR: $arquivod</p>";
    //Mostra o nome do primeiro arquivo agendado e hora para a busca
echo "</p></pre><p>Arquivo do primeiro teste agendado : ag_$arquiv<br />";
echo "</p></pre><p>Acessar este teste através do link ARQUIVO após as: $linharel <br
/></p></pre><p>";
    //Script para reinicializar a rotina do cron
$linhacron = 'ssh gerador@10.24.65.7 crond restart';
    //Abre o arquivo colocando string apartir da ultima linha do arquivo apache
$file= "/var/spool/cron/crontabs/apache";
$fileopen = fopen($file, "a+");
fwrite($fileopen,$linhaqw1 . $linhaG . $linhaqw1 . $linhaSq . $linhaq . $linhaCq . $linhaqq .
$linhaFq . $linhaqq . $linhaGetq . $linhaqq . $linhaGetq1 . $linhaqw2 . $linhaE);
fclose($fileopen);
//Inicializa 2 processos simultaneos com os mesmos passos do processo anterior
if($_POST["dest1"] > '0') {
$linhaSq1 = 'ssh gerador@'. $_POST["dest1"] . ' screen -d -m '. $local[$servidor] . 'ITGRecv -l
/home/basemichele/ag21_$arquiv" . "\n";
$linhaCq1 = 'ssh gerador@'. $_POST["org1"] . ' screen -d -m '. $local[$servidor] . 'ITGSend';
$linhaCq1 .= '-a ' . $_POST["dest1"] ;
    if($_POST["metrica"] == '2') {
        $linhaCq1 .= '-m RTTM';}
    $linhaCq1 .= '-t ' . $_POST["duracao"];

```

```

$linhaCq1 .= '-C ' . $_POST["pacote"];
$linhaCq1 .= '-rp ' . $_POST["portaD"];
$linhaCq1 .= '-sp ' . $_POST["portaO"];
$linhaCq1 .= '-T ' . $_POST["protocolo"];
if($_POST["geracao"] == 'U') {
    $linhaCq1 .= '-u ' . $_POST["valoresU"];
} elseif($_POST["geracao"] == 'P') {
    $linhaCq1 .= '-O ' . $_POST["pctP"];
} else {
    $linhaCq1 .= '-c ' . $_POST["tampac"] . "\n";
}
//Script para terminar processo
$linhaFq1 = 'ssh gerador@'. $_POST["dest1"] . ' killall ITGRecv;';
$linhacron1 .= 'ssh gerador@10.24.65.7 crond restart;';
$linhaGetq1 = 'scp gerador@'. $_POST["dest1"] . " :/home/basemichele/ag21_$arquiv
/home/gerador/basemichele/" . "\n";
$linhaGetq11 = "/home/gerador/basemichele/ITGDec /home/gerador/basemichele/ag21_$arquiv
>ag2_$arquiv" . "\n";
$linhaE1 = "cp /var/www/ag2_$arquiv /home/gerador/basemichele" . "\n";
$linhaFq1 = 'ssh gerador@'. $_POST["dest1"] . ' killall ITGRecv' . "\n";
echo "</p></pre><p>Arquivo do segundo teste agendado : ag2_$arquiv<br />";
echo "</p></pre><p>Acessar este teste através do link ARQUIVO após as: $linharel <br
/></p></pre><p>";
//Script para escrever rotina de cron
$file= "/var/spool/cron/crontabs/apache";
$fileopen = fopen($file, "a+");
fwrite($fileopen,$linhaqw1 . $linhaSq1 . $linhaq . $linhaCq1 . $linhaqq . $linhaFq1 . $linhaqq
.$linhaGetq1 . $linhaqq . $linhaGetq11 . $linhaqw2 . $linhaE1);
fclose($fileopen);
$fname = "/home/gerador/basemichele/ag2_" . $arquiv;}
//Restart linha de cron do arquivo apache executando o arquivo
system ($linhacron);}

```

Perfil Personalizado – Modo Instantâneo

```

if(isset($_POST["submitDITG"])) {
    //Script para servidor e cliente
    $tempo=(($_POST["duracao"])/1000); // ditg tempo em msec. Granularidade foi reduzida a seg.
    $linhaS = 'ssh gerador@'. $_POST["dest"] .' screen -d -m '. $local[$servidor] . 'ITGRecv -l
/home/basemichele/rc_'. $arquivo .';';
    $linhaSh = 'ssh gerador@'. $_POST["dest1"] .' screen -d -m '. $local[$servidor] . 'ITGRecv -l
/home/basemichele/rcd_'. $arquivo .';';
    $linhaC = 'ssh gerador@'. $_POST["org"] .' screen -d -m '. $local[$servidor] .'ITGSend';
        $linhaC .= ' -a ' . $_POST["dest"] ;
        if($_POST["metrica"] == '2') {
            $linhaC .= ' -m RTTM';}
            $linhaC .= ' -t ' . $_POST["duracao"];
            $linhaC .= ' -C ' . $_POST["pacote"];
            $linhaC .= ' -rp ' . $_POST["portaD"];
            $linhaC .= ' -sp ' . $_POST["portaO"];
            $linhaC .= ' -T ' . $_POST["protocolo"];
            if($_POST["geracao"] == 'U') {
                $linhaC .= ' -u ' . $_POST["valoresU"];
            } elseif($_POST["geracao"] == 'P') {
                $linhaC .= ' -O ' . $_POST["pctP"];
            } else {
                $linhaC .= ' -c ' . $_POST["tampac"];}
            $linhaC .= ' -l /home/basemichele/'. $arquivo .';';
            $linhaF = 'ssh gerador@'. $_POST["dest"] .' killall ITGRecv';
            $linhaX='sleep 9;';
        //Script para decodificar arquivo
    $linhaR = 'ssh gerador@'. $_POST["dest"] .' /usr/src/D-ITG-2.6.1b/bin/./ITGDec
/home/basemichele/rc_'. $arquivo .';';
        //Script para copiar relatório no repositório de dados
    $linhaGet = 'scp gerador@'. $_POST["dest"] .'!:/home/basemichele/rc_'. $arquivo.'
/var/www/htdocs/redegiga/imagens/';
        //Script para transformar relatórios em arquivo gráfico

```

```

$linhaE          =          '          /var/www/htdocs/redegiga/imagens/./ITGDec
/var/www/htdocs/redegiga/imagens/rc_'. $arquivo.' -d ' . $tempo.';';

$linhaH = ' /var/www/htdocs/redegiga/imagens/./ITGplot /var/www/htdocs/redegiga/delay.dat
1:4;';

$linhaA          =          '          /var/www/htdocs/redegiga/imagens/./ITGDec
/var/www/htdocs/redegiga/imagens/rc_'. $arquivo.' -j ' . $tempo.';';

$linhaB = ' /var/www/htdocs/redegiga/imagens/./ITGplot /var/www/htdocs/redegiga/jitter.dat
1:4;';

$linhaW          =          '          /var/www/htdocs/redegiga/imagens/./ITGDec
/var/www/htdocs/redegiga/imagens/rc_'. $arquivo.' -p ' . $tempo.';';

$linhaY          =          '          /var/www/htdocs/redegiga/imagens/./ITGplot
/var/www/htdocs/redegiga/packetloss.dat 1:4;';

$linhaWw          =          '          /var/www/htdocs/redegiga/imagens/./ITGDec
/var/www/htdocs/redegiga/imagens/rc_'. $arquivo.' -b ' . $tempo.';';

$linhaYy = ' /var/www/htdocs/redegiga/imagens/./ITGplot /var/www/htdocs/redegiga/bitrate.dat
1:4;';

$linhaCh = 'ssh gerador@'. $_POST["org1"] .' screen -d -m ' . $local[$servidor] .'ITGSend';
    $linhaCh .= ' -a ' . $_POST["dest1"] ;
    $linhaCh .= ' -t ' . $_POST["duracao"];
    $linhaCh .= ' -C ' . $_POST["pacote"];
    $linhaCh .= ' -rp ' . $_POST["portaD"];
    $linhaCh .= ' -sp ' . $_POST["portaO"];
    $linhaCh .= ' -T ' . $_POST["protocolo"];
    if($_POST["metrica"] == '2') {
    $linhaCh .= ' -m RTTM';          }
    if($_POST["geracao"] == 'U') {
    $linhaCh .= ' -u ' . $_POST["valoresU"];
    } elseif($_POST["geracao"] == 'P') {
    $linhaCh .= ' -O ' . $_POST["pctP"];
    } else {
    $linhaCh .= ' -c ' . $_POST["tampac"];
    }
$linhaCh .= ' -l /home/basemichele/l_'. $arquivo .'!';
$linhaFh = 'ssh gerador@'. $_POST["dest1"] .' killall ITGRecv;';
$linhaXh='sleep 9;';

```

```

$linhaXh='sleep '.$stempo.';';
$linhaRh = 'ssh gerador@'. $_POST["dest1"] .' /usr/src/D-ITG-2.6.1b/bin/./ITGDec
/home/basemichele/rcd_'. $arquivo .';';
$linhaGeth = 'scp gerador@'. $_POST["dest1"] .':/home/basemichele/rcd_'. $arquivo.'
/home/gerador/basemichele/';';
$linhaGel = 'cd /home/gerador/basemichele/';';
$linhaEh = './ITGDec rcd_'. $arquivo.' -d ' '.$stempo.';';
$linhaHh = './ITGplot delay.dat 1:4;';
$linhaAh = './ITGDec rcd_'. $arquivo.' -j ' '.$stempo.';';
$linhaBh = './ITGplot jitter.dat 1:4;';
$linhaWh = './ITGDec rcd_'. $arquivo.' -p ' '.$stempo.';';
$linhaYh = './ITGplot packetloss.dat 1:4;';
$linhaWhh = './ITGDec rcd_'. $arquivo.' -b ' '.$stempo.';';
$linhaYhh = './ITGplot bitrate.dat 1:4;';
$arquivod = date ("H:i:s"). "";
echo "</p></pre><p><center>HORA LOCAL INICIAL DO TESTE: $arquivod<center><br
/></p></pre><p>";
echo "</p></pre><p>Arquivo do primeiro sentido de medição: rc_ $arquivo<br />";
echo " _____ ";
echo "<p> </p></pre><p> Relatório do Primeiro fluxo:<br>";
echo " _____ ";
//echo "<p> Linha resultado:<br>";
system($linhaS . $linhaSh . $linhaN1 . $linhaN1h . $linhaC . $linhaN2 . $linhaCh . $linhaN2h .
$linhaXh . $linhaF . $linhaFh . $linhaR );
$arquivodx = date ("H:i:s"). "";
system ($linhaGet . $linhaE . $linhaH . $linhaI . $linhaA . $linhaB . $linhaW . $linhaY .
$linhaWw . $linhaYy );
echo "<p> </p></pre><p> Relatório do Segundo fluxo:<br>";
echo " _____ ";
echo "</p></pre><p>Arquivo do segundo sentido de medição: rcd_ $arquivo<br />";
echo " _____ ";
system($linhaRh . $linhaGeth . $linhaGel . $linhaEh . $linhaHh . $linhaIh . $linhaAh . $linhaBh
.$linhaWh . $linhaYh . $linhaWhh . $linhaYhh);
echo "</p></pre><p><center>HORA LOCAL FINAL DO TESTE: $arquivodx<center><br
/></p></pre><p>";

```

```

echo "</p></pre><p> NTP da máquina de Destino do primeiro fluxo após ser inicializado o
servidor:<br><pre>";
echo file_get_contents("/srv/httpd/ntpdc/ntp1_". $arquivo);
echo "</p></pre><p> NTP da máquina de Origem do primeiro fluxo após ser inicializado o
cliente:<br><pre>";
echo file_get_contents("/srv/httpd/ntpdc/ntp2_". $arquivo);
echo "</p></pre><p> NTP da máquina de Destino do segundo fluxo após ser inicializado o
servidor:<br><pre>";
echo file_get_contents("/srv/httpd/ntpdc/ntp11_". $arquivo);
echo "</p></pre><p> NTP da máquina de Origem do segundo fluxo após ser inicializado o
cliente:<br><pre>";
echo file_get_contents("/srv/httpd/ntpdc/ntp12_". $arquivo);
$fname = "./logs/rcd_". $arquivo;
file_put_contents($fname,$linhaSh . $linhaCh . $linhaXh . $linhaFh . $linhaRh );
$fname = "./logs/rc_". $arquivo;
file_put_contents($fname,$linhaS . $linhaC . $linhaX . $linhaF . $linhaR );
print("<a href='\"bitrate.php'\">==== |GRAFICO 1o Fluxo - Velocidade de Transmissão x tempo|
====</a><p> </p></pre>");
print("<a href='\"packet.php'\">==== |GRAFICO 1o Fluxo - Pacotes Perdidos x tempo|
====</a><p> </p></pre>");
print("<a href='\"delay.php'\">==== |GRAFICO 1o Fluxo - Atraso x tempo| ====</a><p>
</p></pre>");
print("<a href='\"jitter.php'\">==== |GRAFICO 1o Fluxo - Variação do Atraso x tempo|
====</a><p> </p></pre>");
print("<a href='\"bitrate1.php'\">==== |GRAFICO 2o Fluxo - Velocidade de Transmissão x tempo|
====</a><p> </p></pre>");
print("<a href='\"packet1.php'\">==== |GRAFICO 2o Fluxo - Pacotes Perdidos x tempo|
====</a><p> </p></pre>");
print("<a href='\"delay1.php'\">==== |GRAFICO 2o Fluxo - Atraso x tempo| ====</a><p>
</p></pre>");
print("<a href='\"jitter1.php'\">==== |GRAFICO 2o Fluxo - Variação do Atraso x tempo|
====</a><p> </p></pre>");}

```

Perfil Pré-Definido – Modo Instantâneo

```

if(isset($_POST["submitDITGp"])) {
$stempo=(($_POST["duracao"]/1000)); // ditg tempo em mseg. Granularidade foi reduzida a seg.
$linhaS = 'ssh gerador@'. $_POST["dest"] .' screen -d -m '. $local[$servidor] . 'ITGRecv -l
/home/basemichele/rctp_'. $arquivo .';';
$linhaSh = 'ssh gerador@'. $_POST["dest1"] .' screen -d -m '. $local[$servidor] . 'ITGRecv -l
/home/basemichele/rcdtp_'. $arquivo .';';
$linhaC = 'ssh gerador@'. $_POST["org"] .' screen -d -m '. $local[$servidor] .'ITGSend';
    $linhaC .= ' -a ' . $_POST["dest"] ;
    $linhaC .= ' -t ' . $_POST["duracao"];
    $linhaC .= ' -rp ' . $_POST["portaD"];
    if($_POST["metrica"] == '2') {
        $linhaC .= ' -m RTTM';    }
    if($_POST["protocolo"] == '1') {
        $linhaC .= 'DNS';    }
    if($_POST["protocolo"] == '2') {
        $linhaC .= 'Telnet';    }
    if($_POST["protocolo"] == '3') {
        $linhaC .= 'VoIP -x G.711.1'; }
    if($_POST["protocolo"] == '4') {
        $linhaC .= 'VoIP -x G.711.2';}
    if($_POST["protocolo"] == '5') {
        $linhaC .= 'VoIP -x G.723.1';}
    if($_POST["protocolo"] == '6') {
        $linhaC .= 'VoIP -x G.729.2';}
    if($_POST["protocolo"] == '7') {
        $linhaC .= 'VoIP -x G.729.3';}
    if($_POST["protocolo"] == '8') {
        $linhaC .= 'VoIP -x G.711.1-VAD';}
    if($_POST["protocolo"] == '9') {
        $linhaC .= 'VoIP -x G.711.2-VAD';}
    if($_POST["protocolo"] == '10') {
        $linhaC .= 'VoIP -x G.723.1-VAD';}
    if($_POST["protocolo"] == '11') {

```

```

$linhaC .= 'VoIP -x G.729.2-VAD';}
if($_POST["protocolo"] == '12') {
    $linhaC .= 'VoIP -x G.729.3-VAD';}
$linhaC .= '-l /home/basemichele/g_'. $arquivo .';';
$linhaF = 'ssh gerador@'. $_POST["dest"] .' killall ITGRecv;';
$linhaX='sleep 9;';
$linhaR = 'ssh gerador@'. $_POST["dest"] .' /usr/src/D-ITG-2.6.1b/bin/./ITGDec
/home/basemichele/rctp_'. $arquivo .';';
$linhaGet = 'scp gerador@'. $_POST["dest"] .':/home/basemichele/rctp_'. $arquivo.'
/var/www/htdocs/redegiga/im/';';
$linhaE = 'ssh gerador@'. $_POST["dest"] .' /var/www/htdocs/redegiga/imagens/./ITGDec
/var/www/htdocs/redegiga/imagens/rctp_'. $arquivo.' -d ' . $tempo .';';
$linhaH = 'ssh gerador@'. $_POST["dest"] .' /var/www/htdocs/redegiga/imagens/./ITGplot /var/www/htdocs/redegiga/delay.dat
1:4;';
$linhaA = 'ssh gerador@'. $_POST["dest"] .' /var/www/htdocs/redegiga/imagens/./ITGDec
/var/www/htdocs/redegiga/im/rctp_'. $arquivo.' -j ' . $tempo .';';
$linhaB = 'ssh gerador@'. $_POST["dest"] .' /var/www/htdocs/redegiga/imagens/./ITGplot /var/www/htdocs/redegiga/jitter.dat
1:4;';
$linhaW = 'ssh gerador@'. $_POST["dest"] .' /var/www/htdocs/redegiga/imagens/./ITGDec
/var/www/htdocs/redegiga/im/rctp_'. $arquivo.' -p ' . $tempo .';';
$linhaY = 'ssh gerador@'. $_POST["dest"] .' /var/www/htdocs/redegiga/imagens/./ITGplot
/var/www/htdocs/redegiga/packetloss.dat 1:4;';
$linhaWw = 'ssh gerador@'. $_POST["dest"] .' /var/www/htdocs/redegiga/imagens/./ITGDec
/var/www/htdocs/redegiga/im/rctp_'. $arquivo.' -b ' . $tempo .';';
$linhaYy = 'ssh gerador@'. $_POST["dest"] .' /var/www/htdocs/redegiga/imagens/./ITGplot /var/www/htdocs/redegiga/bitrate.dat
1:4;';
$linhaCh = 'ssh gerador@'. $_POST["org1"] .' screen -d -m ' . $local[$servidor] .'ITGSend';
    $linhaCh .= '-a ' . $_POST["dest1"] ;
    $linhaCh .= '-t ' . $_POST["duracao"];
    $linhaCh .= '-rp ' . $_POST["portaD"];
    if($_POST["metrica"] == '2') {
        $linhaCh .= '-m RTTM';    }
    if($_POST["protocolo"] == '1') {
        $linhaCh .= 'DNS';}
if($_POST["protocolo"] == '2') {

```

```

$linhaCh .= 'Telnet';}
if($_POST["protocolo"] == '3') {
    $linhaCh .= 'VoIP -x G.711.1';}
if($_POST["protocolo"] == '4') {
    $linhaCh .= 'VoIP -x G.711.2';}
if($_POST["protocolo"] == '5') {
    $linhaCh .= 'VoIP -x G.723.1';}
if($_POST["protocolo"] == '6') {
    $linhaCh .= 'VoIP -x G.729.2';}
if($_POST["protocolo"] == '7') {
    $linhaCh .= 'VoIP -x G.729.3';}
    if($_POST["protocolo"] == '8') {
        $linhaCh .= 'VoIP -x G.711.1-VAD';}
if($_POST["protocolo"] == '9') {
    $linhaCh .= 'VoIP -x G.711.2-VAD';}
if($_POST["protocolo"] == '10') {
    $linhaCh .= 'VoIP -x G.723.1-VAD'; }
if($_POST["protocolo"] == '11') {
    $linhaCh .= 'VoIP -x G.729.2-VAD'; }
if($_POST["protocolo"] == '12') {
    $linhaCh .= 'VoIP -x G.729.3-VAD'; }
$linhaCh .= '-l /home/basemichele/itp_'. $arquivo .';';
$linhaFh = 'ssh gerador@'. $_POST["dest1"] .' killall ITGRecv;';
$linhaXh='sleep 9;';
$linhaXh='sleep '.$tempo.';';
$linhaRh = 'ssh gerador@'. $_POST["dest1"] .' /usr/src/D-ITG-2.6.1b/bin/./ITGDec
/home/basemichele/rcdtp_'. $arquivo .';';
$linhaGeth = 'scp gerador@'. $_POST["dest1"] .':/home/basemichele/rcdtp_'. $arquivo.'
/home/gerador/basemichele/im2/';';
$linhaGel = 'cd /home/gerador/basemichele/im2/';';
$linhaEh = './ITGDec rcdtp_'. $arquivo.' -d ' . $tempo.';';
$linhaHh = './ITGplot delay.dat 1:4;';
$linhaAh = './ITGDec rcdtp_'. $arquivo.' -j ' . $tempo.';';
$linhaBh = './ITGplot jitter.dat 1:4;';
$linhaWh = './ITGDec rcdtp_'. $arquivo.' -p ' . $tempo.';';

```

```

$linhaYh = './ITGplot packetloss.dat 1:4;';
$linhaWhh = './ITGDec rcdtp_'. $arquivo.' -b ' . $tempo.';';
$linhaYhh = './ITGplot bitrate.dat 1:4;';
$arquivod = date ("H:i:s"). "";
echo "</p></pre><p><center>HORA LOCAL INICIAL DO TESTE: $arquivod<center><br
/></p></pre><p>";
echo "</p></pre><p>Arquivo do primeiro sentido de medição: rctp_ $arquivo<br />";
echo " _____ ";
echo "<p> </p></pre><p> Relatório do Primeiro fluxo:<br>";
echo " _____ ";
//Linha resultado
system($linhaS . $linhaSh . $linhaN1 . $linhaN1h . $linhaC . $linhaN2 . $linhaCh . $linhaN2h .
$linhaXh . $linhaF . $linhaFh . $linhaR );
$arquivodx = date ("H:i:s"). "";
//Linha para fazer gráficos
system ($linhaGet . $linhaE . $linhaH . $linhaI . $linhaA . $linhaB . $linhaW . $linhaY .
$linhaWw . $linhaYy );
echo "<p> </p></pre><p> Relatório do Segundo fluxo:<br>";
echo " _____ ";
echo "</p></pre><p>Arquivo do segundo sentido de medição: rcdtp_ $arquivo<br />";
echo " _____ ";
system($linhaRh . $linhaGeth . $linhaGel . $linhaEh . $linhaHh . $linhaIh . $linhaAh . $linhaBh
.$linhaWh . $linhaYh . $linhaWhh . $linhaYhh);
echo "</p></pre><p><center>HORA LOCAL FINAL DO TESTE: $arquivodx<center><br
/></p></pre><p>";
echo "</p></pre><p> NTP da máquina de Destino do primeiro fluxo após ser inicializado o
servidor:<br><pre>";
echo file_get_contents("/srv/httpd/ntpdc/ntp1tp_". $arquivo);
echo "</p></pre><p> NTP da máquina de Origem do primeiro fluxo após ser inicializado o
cliente:<br><pre>";
echo file_get_contents("/srv/httpd/ntpdc/ntp2tp_". $arquivo);
echo "</p></pre><p> NTP da máquina de Destino do segundo fluxo após ser inicializado o
servidor:<br><pre>";
echo file_get_contents("/srv/httpd/ntpdc/ntp11tp_". $arquivo);

```

```

echo "</p></pre><p> NTP da máquina de Origem do segundo fluxo após ser inicializado o
cliente:<br><pre>";
echo file_get_contents("/srv/http/ntpdc/ntp12tp_".$arquivo);
$fname = "./logs/rcdtp_". $arquivo;
file_put_contents($fname,$linhaSh . $linhaCh . $linhaXh . $linhaFh . $linhaRh );
$fname = "./logs/rctp_". $arquivo;
file_put_contents($fname,$linhaS . $linhaC . $linhaX . $linhaF . $linhaR );
echo " _____</a><p> </p></pre>";
print("<a href=\"bitratepd.php\">=== |GRAFICO 1o Fluxo - Velocidade de Transmissão x
tempo| ===</a><p> </p></pre>");
print("<a href=\"packetpd.php\">=== |GRAFICO 1o Fluxo - Pacotes Perdidos x tempo|
===</a><p> </p></pre>");
print("<a href=\"delaypd.php\">=== |GRAFICO 1o Fluxo - Atraso x tempo| ===</a><p>
</p></pre>");
print("<a href=\"jitterpd.php\">=== |GRAFICO 1o Fluxo - Variação do Atraso x tempo|
===</a><p> </p></pre>");
print("<a href=\"bitrate1pd.php\">=== |GRAFICO 2o Fluxo - Velocidade de Transmissão x
tempo| ===</a><p> </p></pre>");
print("<a href=\"packet1pd.php\">=== |GRAFICO 2o Fluxo - Pacotes Perdidos x tempo|
===</a><p> </p></pre>");
print("<a href=\"delay1pd.php\">=== |GRAFICO 2o Fluxo - Atraso x tempo| ===</a><p>
</p></pre>");
print("<a href=\"jitter1pd.php\">=== |GRAFICO 2o Fluxo - Variação do Atraso x tempo|
===</a><p> </p></pre>");}
Perfil Pré-Definido – Modo Agendado
if(isset($_POST["submitDITG5pd"])) {
$stempo=( $_POST["duracao"]/1000); // ditg tempo em mseg. Granularidade foi reduzida a seg.
$linhaS5 = 'ssh gerador@'. $_POST["dest5"] .' screen -d -m '. $local[$servidor] . 'ITGRecv -l
/home/basemichele/rcipd_'. $arquivo .';';
$linhaC5 = 'ssh gerador@'. $_POST["org5"] .' screen -d -m '. $local[$servidor] .'ITGSend';
$linhaC5 .= ' -a ' . $_POST["dest5"] ;
    $linhaC5 .= ' -t ' . $_POST["duracao"];
    $linhaC5 .= ' -rp ' . $_POST["portaD"];
if($_POST["metrica"] == '2') {
    $linhaC5 .= ' -m RTTM'; }

```

```

    if($_POST["protocolo"] == '1') {
        $linhaC5 .= 'DNS';}
    if($_POST["protocolo"] == '2') {
        $linhaC5 .= 'Telnet';}
    if($_POST["protocolo"] == '3') {
        $linhaC5 .= 'VoIP -x G.711.1';}
    if($_POST["protocolo"] == '4') {
        $linhaC5 .= 'VoIP -x G.711.2';}
    if($_POST["protocolo"] == '5') {
        $linhaC5 .= 'VoIP -x G.723.1';}
    if($_POST["protocolo"] == '6') {
        $linhaC5 .= 'VoIP -x G.729.2';}
    if($_POST["protocolo"] == '7') {
        $linhaC5 .= 'VoIP -x G.729.3';}
    if($_POST["protocolo"] == '8') {
        $linhaC5 .= 'VoIP -x G.711.1-VAD';}
    if($_POST["protocolo"] == '9') {
        $linhaC5 .= 'VoIP -x G.711.2-VAD';}
    if($_POST["protocolo"] == '10') {
        $linhaC5 .= 'VoIP -x G.723.1-VAD';}
    if($_POST["protocolo"] == '11') {
        $linhaC5 .= 'VoIP -x G.729.2-VAD';}
    if($_POST["protocolo"] == '12') {
        $linhaC5 .= 'VoIP -x G.729.3-VAD';}
    $linhaC5 .= '-l /home/basemichele/k_'. $arquivo .';';
    $linhaX5='sleep '.$tempo.';';
    $linhaF5 = 'ssh gerador@'. $_POST["dest5"] .' killall ITGRecv;';
    $linhaR5 = 'ssh gerador@'. $_POST["dest5"] .' /usr/src/D-ITG-2.6.1b/bin/./ITGDec
/home/basemichele/rcipd_'. $arquivo .';';
    $linhaN15 = 'ssh gerador@'. $_POST["dest5"] .' ntpq -c pe > /srv/httpd/ntpdc/ntpipd_'.
$arquivo .';';
    $linhaN25 = 'ssh gerador@'. $_POST["org5"] .' ntpq -c pe > /srv/httpd/ntpdc/ntpipd_'. $arquivo
.';';
    $linhaGet5 = 'scp gerador@'. $_POST["dest5"] .':/home/basemichele/rcipd_'. $arquivo.'
/home/gerador/basemichele/im3/';';

```

```

$linhaGel5 = 'cd /home/gerador/basemichele/im3/';
$linhaE5 = './ITGDec rcipd_'. $arquivo.' -d ' . $tempo.';';
$linhaH5 = './ITGplot delay.dat 1:4;';
$linhaA5 = './ITGDec rcipd_'. $arquivo.' -j ' . $tempo.';';
$linhaB5 = './ITGplot jitter.dat 1:4;';
$linhaW5 = './ITGDec rcipd_'. $arquivo.' -p ' . $tempo.';';
$linhaY5 = './ITGplot packetloss.dat 1:4;';
$linhaWw5 = './ITGDec rcipd_'. $arquivo.' -b ' . $tempo.';';
$linhaYy5 = './ITGplot bitrate.dat 1:4;';
$arquivod = date ("H:i:s"). "";
echo "</p></pre><p><center>HORA LOCAL INICIAL DO TESTE: $arquivod<center><br
/></p></pre><p>";
echo "</p></pre><p>Arquivo do fluxo: rcipd_ $arquivo<br />";
echo " _____";
echo "<p> </p></pre><p> Relatório:<br>";
echo " _____";

//Inicializar servidor e cliente
system($linhaS5 . $linhaN15 . $linhaC5 . $linhaN25 . $linhaX5 . $linhaF5 . $linhaR5 );
$arquivodx = date ("H:i:s"). "";

//Inicializar copia e decodificação
system($linhaGet5 . $linhaGel5 . $linhaE5 . $linhaH5 . $linhaI5 . $linhaA5 . $linhaB5
.$linhaW5 . $linhaY5 . $linhaWw5 . $linhaYy5 );
echo "</p></pre><p><center>HORA LOCAL FINAL DO TESTE: $arquivodx<center><br
/></p></pre><p>";
echo "</p></pre><p> NTP da máquina de Destino após ser inicializado o servidor:<br><pre>";
echo file_get_contents("/srv/httpd/ntpdc/ntpipd_". $arquivo);
echo "</p></pre><p> NTP da máquina de Origem após ser inicializado o cliente:<br><pre>";
echo file_get_contents("/srv/httpd/ntpdc/ntpipd_". $arquivo);
$fname = "./logs/rctp_". $arquivo;
file_put_contents($fname,$linhaS5 . $linhaC5 . $linhaX5 . $linhaF5 . $linhaR5 );
print("<a href=\"bitrate5pd.php\">==== |GRAFICO Velocidade de Transmissão x tempo|
====</a><p> </p></pre>");
print("<a href=\"packet5pd.php\">==== |GRAFICO Pacotes Perdidos x tempo| ====</a><p>
</p></pre>");
print("<a href=\"delay5pd.php\">==== |GRAFICO Atraso x tempo| ====</a><p> </p></pre>");

```

Apêndice C

Relatórios e Processos

Neste apêndice, são descritas as informações obtidas através dos produtos de cada ferramenta.

(a) Ferramenta IPERF - Perfil de Tráfego UDP: É apresentado o intervalo de tempo de cada fluxo separadamente, seguidos de informações do total de Kbytes recebidos e da taxa total de chegada do fluxo. Na linha final é apresentado o total de datagramas que foram enviados, o tempo total do envio de todo o fluxo em segundos, seguido de informações: (a) quantidade total de Kbytes recebidos; (b) taxa total de chegada dos fluxos; (c) *jitter* médio totalizado entre todos os fluxos; (d) quantidade e a porcentagem final de pacotes perdidos; (e) quantidade final de pacotes recebidos.

(b) Ferramenta IPERF – Perfil de Tráfego TCP: É apresentado o intervalo de tempo de cada fluxo separadamente, seguidos de informações do total de Kbytes recebidos, e a taxa total de chegada do fluxo.

(c) Ferramenta DITG – Perfil de Tráfego Personalizado/Pré-Definido: É apresentado: (a) o total de tempo de envio; (b) o total de pacotes na chegada; (c) a taxa mínima, máxima e média do *delay*; (d) a taxa média do *jitter*; (e) a taxa média de recebimento em kbit/s; (f) o número de Bytes recebidos; (g) a taxa média de pacotes recebidos; (h) a quantidade de pacotes perdidos.

A seguir são descritos os processos executados pelo módulo CGT, para a geração de cada produto.

De um modo geral, um produto é elaborado da seguinte forma:

- ✓ Um usuário fornece o cadastro do formulário de acordo com o perfil de tráfego a ser gerado (UDP, TCP, Personalizado ou Agendado), bem como o endereço de origem e de destino.
- ✓ Cada formulário de perfil de tráfego é tratado individualmente pelo sistema.
- ✓ Cada valor atribuído nos campos do formulário é associado à linha de *script* para geração do tráfego, de acordo com a ferramenta utilizada.
- ✓ Cada endereço de origem e destino atribuído nos campos do formulário, é associado a endereços para conexão remota criptografada, utilizando o aplicativo *OpenSSH*. A partir deste ponto, a máquina de origem passa a ser geradora responsável por enviar o tráfego, e a máquina de destino passa a ser medidora, responsável por coletar o tráfego.
- ✓ Todos os processos são feitos remotamente através das máquinas geradora e medidora.

Os produtos são representados de acordo com a ferramenta e o modo utilizado no experimento.

Relatório IPERF – Modo Agendado

Este relatório visa apresentar informações de medições associadas a cada perfil de tráfego TCP e/ou UDP. Estas informações são apresentadas de acordo com a forma original apresentada pela ferramenta.

Este produto é elaborado, de acordo com a ordenação dos eventos descritos a seguir:

- 1) Adiciona os endereços das máquinas, medidora e geradora, a cada processo remoto a ser executado por script: a) *script* para medição; b) *script* para geração do tráfego – associando valores dos campos do formulário, e direcionando o relatório para um arquivo com as identificações “Agend”_”Perfil de tráfego+IPERF”, seguido da marca de tempo na seqüência "Ano-mês-dia_Hora-minuto"; c) *script* de finalização; d) *script* para a cópia do relatório da máquina geradora; e) *script* para envio do arquivo de cron para a máquina geradora; f) *script* para reinicialização da rotina de cron da máquina geradora; g) *script* para reinicialização da rotina de cron local; h) *script* para remoção do arquivo de cron da máquina geradora.
- 2) Coleta informações sobre a hora local.

- 3) Computa tempo para início de medição. Hora da medição = Hora escolhida pelo usuário – 1 minuto.
- 4) Computa tempo para início de geração. Hora da geração = Hora escolhida pelo usuário.
- 5) Computa tempo de finalização da geração e medição do tráfego. Hora da finalização = tempo de duração+hora escolhida pelo usuário.
- 6) Computa tempo para cópia do relatório. Hora da cópia = Hora de finalização+1 (um) minuto.
- 7) Computa tempo de remoção do arquivo de cron da máquina geradora. Hora da Remoção = Hora da cópia + 1 minuto.
- 8) Acrescenta e formata o arquivo utilizado para atender a rotina de cron da máquina geradora, da seguinte forma: c) hora de geração seguida da linha de *script* de geração, direcionando o relatório para um arquivo que recebe um nome de identificação da ferramenta; b) hora de finalização da medição, seguida da linha de *script* de finalização.
- 9) Inicia o *script* para envio do arquivo de cron para a máquina geradora.
- 10) Inicia *script* para reinicialização da rotina de cron da máquina geradora.
- 11) Acrescenta e formata o arquivo utilizado para atender a rotina de cron local, da seguinte forma: a) hora de medição, seguida da linha de *script* de medição; b) hora de finalização da medição, seguida da linha de *script* de finalização; c) hora da cópia, seguida da linha de *script* destinada à cópia do relatório da máquina geradora; d) hora da remoção, seguida da linha de *script* destinada à remoção do arquivo de cron da máquina geradora.
- 12) Inicia *script* para reinicialização da rotina de cron local.
- 13) Apresenta hora local.
- 14) Apresenta nome do relatório parcial para ser buscado na base de dados.
- 15) Apresenta a previsão do horário de busca do relatório parcial.

Relatório DITG – Modo Agendado

Este relatório visa apresentar informações de medições associadas a cada perfil de tráfego personalizado ou pré-definido. Estas informações são apresentadas de acordo com a forma original apresentada pela ferramenta.

Este produto é elaborado, de acordo com a ordenação dos eventos descritos a seguir:

- 1) Adiciona os endereços das máquinas, medidora e geradora, a cada processo remoto a ser executado por script: a) *script* para medição - direcionando o relatório para um arquivo com as identificações “Agend”_”Perfil de tráfego+DITG”, seguido da marca de tempo na

- seqüência "Ano-mês-dia_Hora-minuto"; b) *script* para geração do tráfego - associando valores dos campos do formulário; c) *script* de finalização da medição do tráfego; d) *script* para a cópia do relatório da máquina geradora; e) *script* para decodificação do relatório; f) *script* para envio do arquivo de cron para a máquina geradora; g) *script* para reinicialização da rotina de cron da máquina geradora; h) *script* para reinicialização da rotina de cron local; i) *script* para remoção do arquivo de cron da máquina geradora.
- 2) Coleta informações sobre a hora local.
 - 3) Computa tempo para início de medição. Hora da medição = Hora escolhida pelo usuário – 1 minuto.
 - 4) Computa tempo para início de geração. Hora da geração = Hora escolhida pelo usuário.
 - 5) Computa tempo de finalização da medição do tráfego. Hora da finalização = tempo de duração+hora escolhida pelo usuário.
 - 6) Computa tempo para cópia do relatório. Hora da cópia = Hora de finalização+1 (um) minuto.
 - 7) Computa tempo de decodificação do relatório. Hora da decodificação = Hora da Cópia+1 (um) minuto.
 - 8) Computa tempo de remoção do arquivo de cron da máquina geradora. Hora da Remoção = Hora da cópia + 1 minuto.
 - 9) Acrescenta e formata o arquivo utilizado para atender a rotina de cron da máquina geradora, da seguinte forma: hora de geração seguida da linha de *script* de geração.
 - 10) Inicia *script* para envio do arquivo de cron para a máquina geradora.
 - 11) Inicia *script* para reinicialização da rotina de cron da máquina geradora.
 - 12) Acrescenta e formata o arquivo utilizado para atender a rotina de cron local, da seguinte forma: a) hora da medição, seguida da linha de *script* de medição; b) hora de finalização da medição, seguida da linha de *script* de finalização da medição; c) hora da cópia, seguida da linha de *script* destinada à cópia do relatório da máquina geradora; d) hora da decodificação, seguida da linha de *script* destinada à decodificação do relatório; e) hora da Remoção, seguida da linha de *script* destinada à remoção do arquivo de cron da máquina geradora.
 - 13) Inicia *script* para reinicialização da rotina de cron local.
 - 14) Apresenta hora local.
 - 15) Apresenta nome do relatório parcial para ser buscado na base de dados.
 - 16) Apresenta a previsão do horário de busca do relatório parcial.

Relatório DITG – Modo Instantâneo

Este relatório visa apresentar informações de medições associadas a cada perfil de tráfego personalizado ou pré-definido no modo texto e gráfico, além de informações sobre sincronismo NTP utilizado entre as máquinas.

Este produto é elaborado, de acordo com a ordenação dos eventos descritos a seguir:

- 1) Adiciona os endereços das máquinas, medidora e geradora, a cada processo remoto a ser executado por *script*: a) *script* para medição - direcionando o relatório para um arquivo com as identificações “Inst”_”Perfil de tráfego+DITG”, seguido da marca de tempo na seqüência "Ano-mês-dia_Hora-minuto"; b) *script* de geração do tráfego - associando valores dos campos do formulário; c) *script* de finalização da medição do tráfego; d) *script* para a cópia do relatório da máquina geradora; e) *script* para decodificação do relatório binário para relatório texto; f) *script* para decodificação do relatório binário para relatório gráfico; g) *script* para envio do arquivo de cron para a máquina geradora; h) *script* para obter informações do sincronismo da máquina geradora; i) *script* para obter informações do sincronismo da máquina medidora; j) *script* para reinicialização da rotina de cron da máquina geradora; k) *script* para remoção do arquivo de cron da máquina geradora.
- 2) Coleta informações sobre a hora local.
- 3) Computa tempo para início de geração. Hora da geração = Hora local + 1 minuto.
- 4) Computa tempo de finalização da medição do tráfego. Hora da finalização = tempo de duração+Hora da geração.
- 5) Inicializa em *background* o *script* de medição.
- 6) Acrescenta e formata o arquivo utilizado para atender a rotina de cron da máquina geradora, da seguinte forma: Hora de geração seguida da linha de *script* de geração, e Hora de geração seguida da linha de *script* para obter informações de sincronismo.
- 7) Inicia *script* para reinicialização da rotina de cron da máquina geradora.
- 8) Inicializa *script* de medição.
- 9) Inicia contagem para estado de espera (*sleep*). Tempo de espera = Tempo de Duração+Hora da geração.
- 10) Inicia *script* de finalização da medição do tráfego.
- 11) Inicia em *background* o *script* para obter informações do sincronismo da máquina geradora.
- 12) Inicia em *background* o *script* para remoção do arquivo de cron da máquina geradora.
- 13) Inicia em *background* o *script* para a cópia do relatório da máquina geradora.

- 14) Inicia script para decodificação do relatório binário para relatório texto.
- 15) Inicia script para decodificação do relatório binário para relatório gráfico.
- 16) Apresenta hora do início da geração do tráfego.
- 17) Apresenta relatório integral
- 18) Apresenta nome do relatório parcial para ser buscado na base de dados.

Relatório IPERF – Modo Instantâneo

Este relatório visa apresentar informações de medições associadas a cada perfil de tráfego UDP ou TCP, além de informações sobre sincronismo NTP utilizado entre as máquinas. Estas informações são apresentadas de acordo com a forma original.

Este produto é elaborado, de acordo com a ordenação dos eventos descritos a seguir:

- 1) Adiciona os endereços das máquinas, medidora e geradora, a cada processo remoto a ser executado por *script*: a) *script* para medição; b) *script* de geração do tráfego - associando valores dos campos do formulário, direcionando o relatório para um arquivo com as identificações “Inst”_”Perfil de tráfego+IPERF”, seguido da marca de tempo na seqüência "Ano-mês-dia_Hora-minuto"; c) *script* de finalização da medição do tráfego; d) *script* de finalização da geração do tráfego; e) *script* para a cópia do relatório da máquina geradora; f) *script* para envio do arquivo de cron para a máquina geradora; g) *script* para obter informações do sincronismo da máquina geradora; h) *script* para obter informações do sincronismo da máquina medidora; i) *script* para reinicialização da rotina de cron da máquina geradora; j) *script* para remoção do arquivo de cron da máquina geradora.
- 2) Coleta informações sobre a hora local.
- 3) Computa tempo para início de geração. Hora da geração = Hora local + 1 minuto.
- 4) Computa tempo de finalização da medição e geração do tráfego. Hora da finalização = Tempo de Duração+Hora da geração.
- 5) Inicia em background o script de medição.
- 6) Acrescenta e formata o arquivo utilizado para atender a rotina de cron da máquina geradora, da seguinte forma: a) Hora de geração seguida da linha de *script* de geração; b) Hora de geração seguida da linha de *script* para obter informações de sincronismo c) Hora da finalização, seguida do *script* de finalização da geração do tráfego.
- 7) Inicia *script* para reinicialização da rotina de cron da máquina geradora.
- 8) Inicializa *script* de medição.

- 9) Inicia contagem para estado de espera (sleep). Tempo de espera = Tempo de Duração+Hora da geração.
- 10) Inicia script de finalização da medição do tráfego.
- 11) Inicia em background o *script* para remoção do arquivo de cron da máquina geradora.
- 12) Inicia em background o *script* para a cópia do relatório da máquina geradora.
- 13) Apresenta relatório integral
- 14) Apresenta nome do relatório parcial para ser buscado na base de dados.

Apêndice D

Pré-Requisitos

Para o funcionamento do SCGT, inicialmente é necessário que a máquina possua o sistema Unix, e tenha uma configuração necessária para o funcionamento, é apresentada a seguir a configuração mínima exigida para a instalação do SCGT:

- Processador Intel Celeron 1.8 GHz.
- 256 MBytes de Memória.
- 1 HD de 40 Gbytes.
- 1 Interface Ethernet UTP 10/100.

Instalação do SCGT

Apesar de estar aqui descrito como instalar o SCGT, este já se encontra instalado em um servidor localizado na UFF (Universidade Federal Fluminense) sob administração do projeto GIGA. Porém como esta é a primeira versão do sistema, futuros trabalhos podem ser realizados para a sua melhoria. Desta forma, são apresentados a seguir os passos a serem feitos para a instalação do sistema.

- 1) Criar grupo *gigaiface*, para acesso e manipulação de arquivos, com `gid=900`. Para o sistema operacional linux *slackware*, a criação pode ser feita através da linha de comando: `“groupadd -g 900 gigaiface”`.
- 2) Criar usuário o usuário `“apache”`. Caso já exista este usuário, apenas o inclua no grupo `“gigaiface”`, criado no passo anterior. Para o sistema operacional *Linux slackware*, a criação pode ser feita através da linha de comando: `“useradd -u 9000 -g 900 -d /var/www/ -s /bin/bash apache”`.
- 3) Configurar o servidor *web* para rodar com o usuário `“Apache”` e grupo `“gigaiface”` criados acima. Para o sistema operacional *Linux slackware*, a configuração pode ser feito pelo arquivo: `“/etc/apache/httpd.conf”`.
- 4) Reiniciar o servidor *Web*.
- 5) Trocar o dono do diretório (e subdiretórios) `“/var/www”` para usuário o `“Apache”`. Para o sistema operacional *Linux slackware*, a criação pode ser feita através da linha de comando: `“cd /var/www; chown -R apache.gigaiface”`.
- 6) Gerar par de chaves públicas e privadas sem senha para acesso remoto. Para o sistema operacional linux *slackware*, a criação pode ser feita através da linha de comando: `“su -apache”`, seguido de `“ssh-keygen -t rsa”`.
- 7) Instalar o aplicativo *Octave* [OCT08], de acordo com recomendações do próprio fabricante.
- 8) Instalar o PHP [PHP08].
- 9) Ativar o servidor *Web*.
- 10) Criar diretório `“/home/gerador/base”` para guardar arquivos de logs. Este diretório deve conter sempre, no mínimo 20 Gbytes de espaço vazio. Para o sistema operacional *Linux slackware*, a criação pode ser feita através da linha de comando: `“cd /home”` seguida de `“mkdir gerador”` e `“mkdir base”`.
- 11) Configurar o arquivo de NTP. Para o sistema operacional *Linux slackware*, a configuração é feita através do diretório `“/etc/ntp.conf”`, e deve ter as seguintes linhas:
 - a. `restrict default noquery notrust nomodify`
 - b. `restrict 127.0.0.1`
 - c. `restrict 10.24.65.0 mask 255.255.255.0`
 - d. `server 10.24.65.8`
 - e. `driftfile /etc/ntp.drift`
 - f. `logfile /var/log/ntp.log`
 - g. `multicastclient`
 - h. `broadcastdelay 0.008`

12) Inicializar o servidor NTP, através da linha de comando “service ntpd start”.

Para o funcionamento do SCGT, inicialmente é necessário que as máquinas de origem e destino possuam o sistema Unix, e tenham o mínimo de configuração necessária para o funcionamento. A seguir é indicada a configuração mínima exigida para a instalação das máquinas:

- Processador Intel Celeron 1.8 GHz.
- 256 MBytes de Memória.
- 1 HD de 20 Gbytes.
- 1 Interface Ethernet UTP 10/100.

Instalações nas Máquinas de Origem e Destino

Cada ferramenta de geração e medição de tráfego deve ser instalada, dependendo do perfil de tráfego a ser gerado, podendo apenas ser instalada uma única ferramenta para um par de máquinas. A seguir são detalhadas as instruções para a configuração de cada máquina onde deseja gerar/e ou coletar o tráfego.

- 1) Criar grupo gigaiface, para acesso e manipulação de arquivos, com gid=900. Para o sistema operacional linux *slackware*, a criação pode ser feita através da linha de comando: “groupadd -g 900 gigaiface”.
- 2) Criar usuário o usuário “gerador”. Caso já exista este usuário, apenas o inclua no grupo “gigaiface”, criado no passo anterior. Para o sistema operacional *Linux slackware*, a criação pode ser feita através da linha de comando: “useradd -o -u 0 -g 900 -d /usr/local/gerador -s /bin/bash gerador”.
- 3) Destrua a senha do usuário gerador, modificando-a para um valor qualquer.
- 4) Criar diretório “/home/gerador/” para guardar arquivos de logs. Este diretório deve conter sempre, no mínimo 5 Gbytes de espaço vazio. Para o sistema operacional *Linux slackware*, a criação pode ser feita através da linha de comando: “cd /home” seguida de “mkdir gerador”.
- 5) Baixar do servidor SCGT através do link <http://200.20.0.151/redegiga/im/>, a chave pública do usuário “controlador”, e instalar esta para o usuário “gerador”. Somente deste modo será possível o login remoto pelo sistema SCGT. Para o sistema operacional *Linux slackware*, a criação pode ser feita através da linha de comando: “cat apache-controlador-id_rsa.pub > ~gerador/.ssh/authorized_hosts”.

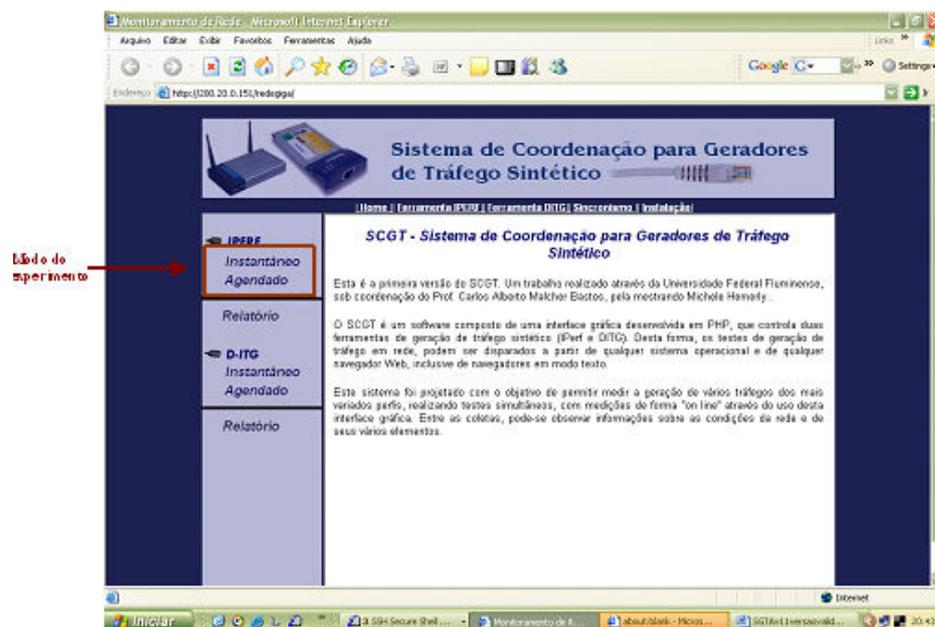
- 6) Instalar a ferramenta D-ITG [DIT08] no diretório “/usr/src/”, de acordo com recomendações do próprio fabricante.
- 7) Instalar a ferramenta IPERF [DIT08] no diretório “/usr/src/”, de acordo com recomendações do próprio fabricante.
- 8) Criar usuário o usuário “ditg”.
- 9) Criar usuário o usuário “IPERF”.
- 10) Criar usuário o usuário “apache”.
- 11) Configurar o arquivo de NTP. Para o sistema operacional linux slackware, a configuração é feita através do diretório “/etc/ntp.conf”, e deve ter as seguintes linhas:
 - restrict default noquery notrust nomodify
 - restrict 127.0.0.1
 - restrict 10.24.65.0 mask 255.255.255.0
 - server 10.24.65.8
 - driftfile /etc/ntp.drift
 - logfile /var/log/ntp.log
 - multicastclient
 - broadcastdelay 0.008
- 12) Inicializar o servidor NTP.

Apêndice E

Operacionalidade

Este apêndice expõe a operacionalidade dos diferentes modos de geração do tráfego do SCGT através da utilização na *Web*. Veja na Figura 31 a apresentação da interface gráfica inicial.

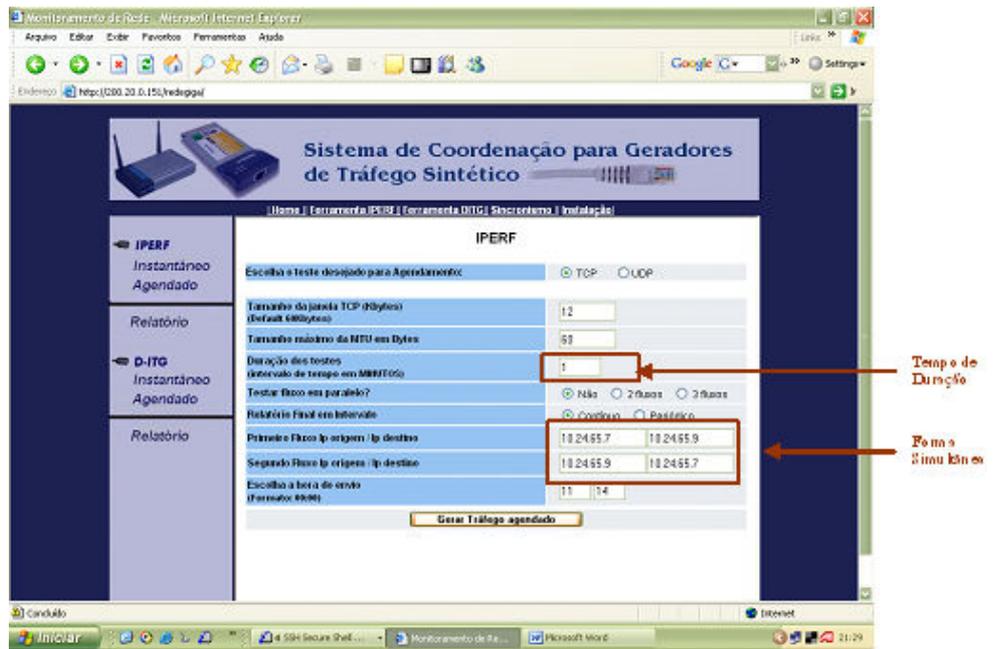
Figura31: Apresentação do SCGT



Ao clicar em cada modo de experimento, automaticamente o sistema direciona para uma página que contém formulários para compor o perfil de tráfego desejado para o experimento, bem como a forma como o mesmo será gerado (simultâneo ou não).

Se for preenchido apenas um único endereço para a origem e o destino, o tráfego é inicializado para apenas este fluxo, e isto significa um tráfego não simultâneo. Porém, se for preenchido o endereço origem e destino para os dois fluxos, o tráfego é inicializado para estes simultaneamente. Veja na Figura 32, um exemplo do formulário da ferramenta IPERF no modo agendado, utilizando o perfil de tráfego TCP na forma simultânea.

Figura 32: Perfil de Tráfego- TCP – Modo Agendado



O tempo de recebimento do relatório é indiretamente estabelecido pelo modo do serviço a ser executado (instantâneo ou agendado).

O resultado na Figura 33 obtido através do *Link* de Relatório da ferramenta IPERF:

Figura 33: Lista de Relatórios

