PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE ENGENHARIA ELÉTRICA

VINÍCIUS TAVARES GUIMARÃES

# AMOSTRAGEM ALEATÓRIA ESTRATIFICADA ADAPTATIVA PARA IDENTIFICAÇÃO DE FLUXOS "ELEFANTES" EM REDES CONVERGENTES

Porto Alegre
2007

VINÍCIUS TAVARES GUIMARÃES

# AMOSTRAGEM ALEATÓRIA ESTRATIFICADA ADAPTATIVA PARA IDENTIFICAÇÃO DE FLUXOS "ELEFANTES" EM REDES CONVERGENTES

Dissertação apresentada como requisito para obtenção do grau de Mestre, pelo Programa de Pós-graduação da Faculdade de Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Dr. Fabian Vargas

Porto Alegre
2007

Dedico esta dissertação aos meus pais Daniel Silva Guimarães e Ana Leonor Tavares Guimarães, aos meus irmãos Daniel Silva Guimarães Jr. e Gabriel Tavares Guimarães e a minha noiva Fernanda Barreto Mielke, os quais foram indispensáveis nesta caminhada. Adicionalmente, aos amigos e colegas do GPARC&TI.

## AGRADECIMENTOS

Inicialmente, quero agradecer ao professor Dr. Jorge Guedes, pelo apoio, atenção e incentivo dispensado durante a realização do mestrado. Gostaria também de expressar minha gratidão ao professor Dr. Fabian Vargas por ter aceitado o convite para ser o orientador do trabalho e se mostrar sempre a disposição.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) por ter proporcionado a realização do mestrado, através da concessão da bolsa de estudo.

Aos colegas e amigos do GPARC&TI. Em especial aos amigos Roberto Costa, Gléderson Santos, Mateus Caruccio, Ricardo Balbinot e Eloísio Bergamaschi, pelo auxílio efetivo no desenvolvimento da dissertação, fato que, indiscutivelmente, elevou significativamente a qualidade do trabalho.

Aos professores e funcionários do Programa de Pós-Graduação em Engenharia Elétrica, os quais sempre se apresentaram acessíveis e presentes nos momentos de auxílio.

Agradeço a todos meus amigos e amigas, os quais foram extremamente importantes nesta árdua caminhada. Não irei citar nomes para não ser injusto, no caso de esquecer de alguém.

A minha noiva e amiga Fernanda, a qual foi indispensável em todos os momentos.

Aos meus familiares, pelo carinho, estímulo e compreensão, em especial aos meus pais Daniel Silva Guimarães e Ana Leonor Tavares Guimarães, aos meus irmãos Daniel Silva Guimarães Jr. e Gabriel Tavares Guimarães, por todo apoio, companheirismo e dedicação. Ao meu pai faço um agradecimento adicional, pois além do apoio natural de um pai, contribui com sua experiência acadêmica e com seu conhecimento nas áreas de matemática e estatística.

Finalmente, mas não em último lugar, agradeço a **DEUS** por mais esta oportunidade de crescimento pessoal.

“Há homens que lutam um dia e são bons. Há outros que lutam um ano e são melhores. Há os que lutam muitos anos e são muito bons. Porém, há os que lutam toda a vida. Esses são os imprescindíveis.” *Bertolt Brecht*

## RESUMO

Técnica de amostragem aleatória estratificada adaptativa aplicada à identificação de grandes fluxos (fluxos "Elefante"), no contexto das redes de comunicação convergentes baseadas no modelo IP foi implementada, avaliada e os resultados obtidos confrontados com os obtidos em um sistema tradicional de medição de fluxos. Foi, ainda, efetuado o diagnóstico das correlações e divergências das informações inferidas com respeito à precisão, confiabilidade e ocorrência de falsos positivos e falsos negativos. Mostrou-se que a técnica de amostragem aleatória estratificada adaptativa requer o uso de mecanismos especificamente desenvolvidos para a sua utilização e deve ser empregada com base em um conhecimento prévio do comportamento usual da rede. Verificou-se que o erro percentual, no uso da técnica de amostragem aleatória estratificada adaptativa, para fluxos considerados "elefante", não ultrapassa 3% nas estimativas de contabilização de pacotes e volume de bytes; que o modelo temporal AR(1) para cinco valores passados faz com que o ajuste da taxa de amostragem seja efetivamente adaptativo e que, para condições de tráfego com oscilações drásticas, o modelo temporal AR(1) com três valores passados apresenta uma convergência maior que o modelo AR(1) para cinco e sete valores passados. É, ainda, apresentada uma revisão bibliográfica abrangendo os principais aspectos relacionados ao gerenciamento de redes, convergindo ao estado da arte relacionado à aplicação da amostragem de pacotes. Adicionalmente, é apresentado o delineamento da técnica de amostragem abordada no estudo, sua implementação e as principais discussões acerca dos resultados obtidos.



Palavras-chave: Medição e monitoramento de redes IP. Medição Passiva. Amostragem aleatória estratificada. Estatísticas de fluxos. Fluxos "Elefante".

**ABSTRACT**

Adaptive stratified random packet sampling technique to identify large flows (“Elephant” flows) in the context of the convergent communication networks based on the IP model was implemented, evaluated and the obtained results compared with the results collected from traditional *per-flow* measurement system. The correlations and divergences diagnosis of the inferred information about precision, reliability and occurrence of false positive and false negative, also, was made. It was shown that the adaptive stratified random sampling requests the use of mechanisms specifically developed and it should be used with base in a previous knowledge of the usual network behavior. It was verified that, using the adaptive stratified random sampling technique, the percentile error for "elephant" flows was less than 3% in the estimation of packages and volume of bytes account; that the time model AR(1) for five past values makes the sampling technique truly adaptive and, for bursty traffic conditions, the time model AR(1) for three past values presents a larger convergence than the model AIR (1) for five or seven past values. This work also shows a bibliography review of the main aspects related to network management, converging to the state of art related to the application of the sampling packets technique. Additionally, the used sampling technique is presented and results achieved are discussed.

Keywords: Monitoring and measurements on IP networks. Passive Measurements. Stratified Random Sampling. Flows statistics. Elephant flows.

## LISTA DE ILUSTRAÇÕES

## LISTA DE TABELAS

## LISTA DE SÍMBOLOS

$m^f$ - Os pacotes pertencente ao fluxo $f$ dentre o total de pacotes $m$

$p^f$ - A proporção da contabilização de pacotes para o fluxo $f$

$p^\theta$ - Limiar pré-estabelecido utilizado na identificação dos fluxos "elefantes"

$\hat{m}^f$ - Estimativa do total de pacotes para o fluxo $f$

$\hat{v}^f$ - Estimativa do total de *bytes* para o fluxo $f$

$p^f$ - Proporção de pacotes para o fluxo $f$

$\hat{p}^f$ - Proporção de amostragem para o fluxo $f$

$Y_p \sim N(0,1)$ - Função de densidade de probabilidade padronizada

$\Phi(\cdot)$ - Função de Distribuição Cumulativa

$n^{*,p}$ - Número mínimo de amostras requeridas para estimação do total de pacotes

$\mu^f$ - Média do tamanho dos pacotes pertencentes ao fluxo $f$

$S^f$ - SCV, Quadrado do Coeficiente de Variação do tamanho dos pacotes pertencentes ao fluxo $f$

$S^\theta$ - SCV para fluxos "elefantes"

$n^{*,b}$ - Número mínimo de amostras requeridas para estimação do volume de *bytes*

$m_h$ - Total de pacotes pertencentes ao bloco $h$

$\hat{m}_h^f$ - Estimativa do total de pacotes para o fluxo $f$, no bloco $h$

$\hat{v}_h^f$ - Estimativa do volume de *bytes* para o fluxo $f$, no bloco $h$

$\eta$ - Nível de significância

$\varepsilon$ - Valor máximo, em módulo, do erro relativo (precisão)

## ACRÔNIMOS

| | |
|---|---|
| 3GPP - | *Third Generation Partnership Project* |
| AF - | *Assured Forwarding* |
| AMP - | *Active Measurement Project* |
| API - | *Application Programming Interface* |
| AQUILA - | *Adaptive Resource Control for QoS Using an IP-based Layered Architecture* |
| AR - | *Auto-regressive* |
| BBS - | *Biased Systematic Sampling* |
| BPF - | *Berkeley Packet Filter* |
| CA - | *Computer Associates* |
| DDoS - | *Distributed denial-of-service* |
| DTD - | *Document Type Definition* |
| ER - | Entidade-Relacionamento |
| FIFO - | *First-In-First-Out* |
| FTP - | *File Transfer Protocol* |
| GPARC & TI - | Grupo de Pesquisas Avançadas em Redes de Comunicação e Tecnologia da Informação |
| GPS - | *Global Positioning System* |
| HP - | *Hewlett-Packard* |
| HTTP - | *Hypertext Transfer Protocol* |
| IBM – | *International Business Machine* |
| ICMP - | *Internet Control Message Protocol* |
| IETF - | *Internet Engineering Task Force* |
| IP - | *Internet Protocol* |
| IPFIX - | *IP Flow Information Export* |
| IPPM - | *IP Performance Metrics* |
| ITU - | *International Telecommunication Union* |

MIB - *Management Information Base*

MTU - *Maximum Transmission Unit*

MVCC - *Multi-Version Concurrency Control*

NeMaC - *Network Manager Collector*

NeTraMet - *Network Traffic Meter*

NLANR - *National Laboratory for Applied Network Research*

NMS - *Network Management System*

PMA - *Passive Measurement and Analysis*

POO - Programação Orientada a Objetos

POP - *Point of Presence*

PoP-PE - Ponto de Presença de Pernanbuco

PSAMP - *Packet Sampling*

QoS - *Quality of Service*

RFC - *Request For Comment*

RMON - *Remote Network Monitoring*

RMONMIB - *Remote Network Monitoring*

RNG - *Random Number Generation*

RNP - Rede Nacional de Pesquisa

RTFM - *Realtime Traffic Flow Measurement*

RTP - *Real Time Protocol*

SCV - *Squared Coeficient of Variation*

SFTP - *Secure File Transfer Protocol*

SGBD - Sistema Gerenciador de Banco de Dados

SLA - *Service Level Agreement*

SLM - *Sevice Level Management*

SLS - *Service Level Specification*

SNMP - *Simple Network Management Protocol*

SO - Sistema Operacional

SQL - *Structured Query Language*

SSL - *Secure Sockets Layer*

STL - *Standard Template Library*

TCP - *Transmission Control Protocol*

TEQUILA - *Traffic Engineering for Quality of service in the Internet at Large*

TEWG - *Internet Traffic Engineering*

TEWG - *Traffic Enginneering working group*

TI - Tecnologia da Informação

TIC - Tecnologias da Informação e Comunicação

TLC - Teorema do Limite Central

ToS - *Type of Service*

UAMA - *Universal Active Measurement Architecture*

UML - *Unified Modeling Language*

VLL - *Virtual Leased Line*

XML - *Extended Markup Language*

**SUMÁRIO**

# 1 INTRODUÇÃO

Os esforços da comunidade acadêmico-científica no contexto de redes de comutação de pacotes fundamentadas no modelo IP têm sido, na última década, direcionados fortemente para as demandas e necessidades relacionadas aos sistemas de gerenciamento. Atualmente, com o advento das redes convergentes e a conseqüente possibilidade de oferta de diferentes tipos de serviços, este nicho de atuação torna-se ainda mais fértil e instigante, pois cada vez mais se materializa como vital para as organizações/corporações o gerenciamento pleno, efetivo e, desejavelmente, proativo dos ambientes de rede.

Neste sentido, *Xu et. al.* [XU 05] enfatiza que a medição e monitoramento efetivo do tráfego são indispensáveis para o gerenciamento de *Quality of Service* (QoS), planejamento de recursos e projeto de infra-estrutura da rede. Adicionalmente, *Estan* e *Varghese* [EST 04] destacam que as informações de medição são essenciais para o monitoramento em curto prazo (identificação de ataques por negação de serviços), engenharia de tráfego em longo prazo (alternâncias no roteamento do tráfego) e contabilização (para fixar tarifas com base no uso dos recursos).

Sob esta perspectiva, duas técnicas de medição destacam-se na obtenção de dados de desempenho da rede: medição passiva e medição ativa. A medição ativa propõe-se a obter indicadores de desempenho a partir da transmissão controlada de tráfego de teste através da rede que se deseja analisar. Segundo [TRI 02], os dados obtidos a partir da medição ativa provêem as seguintes informações relacionadas à rede: topologia, largura de banda disponível e no gargalo, atraso em uma via, atraso de ida e volta, perda, variação no atraso (*jitter*) e grau de desordem dos pacotes.

De outro lado, a premissa básica da técnica de medição passiva está centrada no fato de que esta não requer a injeção de tráfego adicional para mensuração da rede operacional. Dessa forma, o tráfego habitual pode ser monitorado com a adição de um ou mais pontos de medição em um determinado domínio de rede. Dentre as principais informações coletadas a partir desta técnica, pode-se destacar a classificação e contabilização do tráfego baseada em fluxos. Particularmente, a medição em nível de fluxos provê informações para aplicações de análise

do perfil de tráfego, matriz de tráfego, monitoramento de QoS, contabilização baseada em uso, entre outras.

Entretanto, com o aumento na capacidade das tecnologias de transmissão em rede (atualmente na ordem de 100 *Mbps* – 10 *Gbps*), a classificação e quantificação do tráfego em fluxos, através do processamento de todos os pacotes que transpassam o ponto de observação, acabam tornando-se proibitivas. Segundo [CHO 04], as informações de fluxo são tipicamente armazenadas em *software* e a capacidade de processamento não acompanha a capacidade de transmissão dos enlaces. Além disso, a análise de todos os pacotes contemplando a atualização, armazenamento e exportação das informações de fluxo, requer um alto poder de processamento, capacidade de memória *cache*, requisições de *I/O* e consumo de largura de banda.

Partindo deste cenário, observa-se, claramente, que a medição de tráfego baseada em fluxos apresenta como problema principal a falta de escalabilidade. Estudos realizados no final da década de noventa [THO 97][FAN 99], os quais utilizaram *traces* de diferentes *backbones*, já indicavam que o número de fluxos entre pares de *hosts* no período de uma hora atingia a marca de 1,7 milhões. Na atualidade, é factível inferir que estes valores cresceram drasticamente.

Endereçando estas restrições, a utilização de mecanismos para amostragem de pacotes apresenta-se no cenário de pesquisa como uma aproximação capaz de solucionar os problemas acima apresentados. No *Internet Engineering Task Force* (IETF), os grupos de trabalho *PSAMP* [QUI 06] e *IPFIX* [PLO 06] recomendam fortemente em suas especificações o uso de amostragem de pacotes. A *Request For Comment* (RFC) 3176 [PHA 01] apresenta o *sFlow* como tecnologia embarcada em roteadores e *switches* para monitoramento do tráfego utilizando amostragem. Além disso, diferentes fabricantes como *Juniper* e *Cisco* já implementam em seus dispositivos métodos estáticos de amostragem, como por exemplo: "amostrar 1 em *N*".

Trabalhos atuais [MOR 04][CHO 04][CHO 06][EST 03] especificam ainda mais o escopo de aplicação dos mecanismos de amostragem, focando apenas em grandes fluxos ou fluxos "elefantes". O direcionamento para este tipo de fluxo é justificado por diferentes estudos apresentados em [ZHA 02][FEL 01], os quais demonstram a prevalência do fenômeno

do “rato” e do “elefante” para fluxos definidos em diferentes níveis de granularidade: um pequeno percentual dos fluxos, tipicamente, contabiliza para um grande percentual do tráfego total. Em [MOR 04] é mostrado que em um dos *traces* analisados, cerca de 0.02% do total de fluxos contribuíram em mais de 59% do volume total do tráfego. Desta forma, para muitas aplicações de monitoramento e medição, prover estatísticas acuradas apenas dos fluxos elefantes é suficiente.

Ainda neste contexto, em [EST 03] é apresentada uma citação bastante interessante, a qual sintetiza uma das principais justificativas para o foco em fluxos “elefantes”:

> Para manter-se o estado de cada fluxo, tem-se um problema de escalabidade e estar-se-á rastreando milhões de formigas para rastrear alguns elefantes. *Van Jacobson, End-to-end Research meeting*, Junho de 2000.

Observando esta tendência com relação a medição de tráfego baseada em fluxos e reconhecendo as demandas atuais dos sistemas de medição desenvolvidos pelo Grupo de Pesquisas Avançadas em Redes de Comunicação e Tecnologia da Informação (GPARC&TI), o presente trabalho apresenta a implementação e avaliação da técnica de amostragem aleatória estratificada adaptativa na identificação de fluxos “elefantes”, proposta por [CHO 04][CHO06].

## 1.1 OBJETIVOS

A partir da contextualização previamente apresentada e em conformidade com as motivações relacionadas ao desenvolvimento do presente trabalho, são delineados os seguintes objetivos:

- implementar a técnica de amostragem aleatória estratificada adaptativa proposta por *Choi et. al.* [CHO 04][CHO 06], para a identificação de grande fluxos (fluxos “*Elephant*”) em redes convergentes como módulo adicional ao sistema tradicional para a medição de tráfego baseada em fluxos, desenvolvido pelo GPARC&TI;

- avaliar e comparar os resultados obtidos confrontando-os com os resultados coletados a partir do sistema tradicional para a medição de tráfego baseada em fluxos.

## 1.2 ORGANIZAÇÃO DA DISSERTAÇÃO

A estruturação desta dissertação foi concebida com o intuito de apresentar de forma clara o contexto global no qual a proposta se enquadra, aprofundando os tópicos específicos no decorrer do texto. Para tal, utilizou-se o modelo de dissertação disponibilizado no *site* da biblioteca da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS). Além da introdução, o trabalho conta ainda como outros cinco capítulos.

O capítulo 2 apresenta a revisão de literatura relacionada ao escopo no qual o trabalho se enquadra. Inicialmente, é apresentada uma contextualização dos principais conceitos relacionados ao gerenciamento e monitoramento de redes, destacando a obtenção de indicadores de desempenho, em especial, a medição de tráfego baseada em fluxos. Como fechamento, é apresentado o estado da arte em relação à aplicação de técnicas de amostragem no contexto de gerenciamento de redes, focando nas proposições relacionadas à medição baseada em fluxos, especificamente, na identificação de grandes fluxos (fluxos "*Elephant*").

No capítulo 3 é apresentada a técnica de amostragem aleatória estratificada adaptativa. Para tal, são apresentados os conceitos relacionados à identificação de fluxos "elefantes" no contexto desta técnica de amostragem. Após, é mostrado o delineamento estatístico utilizado pelos autores para definição do número de amostras requeridas em cada estrato, com base em um nível de tolerância de erro pré-estabelecido. Por fim, é apresentada a aproximação de predição utilizada para ajustar a taxa de amostragem.

O capítulo 4 apresenta as tecnologias utilizadas no desenvolvimento do trabalho e os ambientes de teste utilizados (ilustrando os modelos dos cenários de rede, condições de contorno e operação, método empregado e resultados esperados). Além disso, são

apresentadas as particularidades relacionadas ao desenvolvimento propriamente dito do protótipo.

Os resultados e discussões relacionadas ao desenvolvimento do trabalho são apresentados no capítulo 5. Inicialmente, são explicitadas considerações que permeiam a implementação propriamente dita, destacando como foram resolvidos os principais pontos de incerteza. Adicionalmente, são apresentados os resultados das avaliações realizadas a partir do ambiente de teste proposto. Especificamente, é realizada a avaliação do processo de predição utilizando o Modelo AR(1), além da averiguação das estimativas para o total de pacotes e volume de *bytes*, resultantes do uso da técnica de amostragem.

As conclusões obtidas a partir do trabalho desenvolvido podem ser encontradas no capítulo 6 . Além disso, são apresentadas algumas propostas de trabalhos futuros a cerca do tema, tanto em relação à continuidade do protótipo desenvolvido, como ao desenvolvimento de avaliações específicas relacionadas a outras técnicas de amostragem disponíveis, direcionando, em particular, para a possibilidade de proposição de uma nova técnica.

# 2 REVISÃO BIBLIOGRÁFICA

O presente capítulo tem como objetivo principal apresentar uma revisão dos principais conceitos, técnicas, padrões e iniciativas utilizadas na medição e monitoramento de redes IP, focando na medição de tráfego baseado em fluxos. Desta forma, buscou-se contextualizar o cenário no qual este trabalho se enquadra, direcionando-o para o atual estado da arte.

## 2.1 GERENCIAMENTOS DE REDES IP

O gerenciamento das redes de comunicação baseadas no modelo IP, especificamente as redes de computadores, foi inicialmente impulsionado pela necessidade de monitoramento e controle do universo de dispositivos que as constituem. Atualmente, tais redes e seus recursos associados, além das aplicações distribuídas, têm se tornado de importância fundamental e determinantes para as organizações.

Segundo [GAS 01], a utilização das redes de computadores como suporte para um crescente número de negócios e aplicações críticas tem estimulado a busca de soluções de gerenciamento que permitam manter em funcionamento não apenas a infra-estrutura física da rede, mas também os protocolos e serviços que a compõem. Neste sentido, a responsabilidade dos gerentes de rede é cada vez maior, exigindo, imperativamente, a utilização de ferramentas automatizadas.

Com base neste apelo, a expansão do mercado de *softwares* para gerenciamento de rede está crescendo de forma bastante consistente, propulsando as empresas do ramo nos diferentes segmentos do mercado. Segundo [VIE 04], dentre a gama de soluções possíveis para o gerenciamento de redes, uma das mais usuais consiste em utilizar um computador que interage com os diversos componentes da rede para extrair dele as informações necessárias ao seu gerenciamento.

Obviamente, outras premissas também constituem os requisitos para que um sistema de gerenciamento obtenha os resultados esperados e possa ser efetivo no auxílio aos gerentes de rede. No sentido de garantia de alta disponibilidade e eficiência da rede de comunicação, é evidente a necessidade de um ambiente de gerenciamento flexível que possa prover agilidade de adaptação a cenários dinâmicos.

Em [GAS 01], é destacado que a flexibilidade de um sistema robusto de gerenciamento também deve considerar o crescente aumento no tamanho das redes, requerendo muitas vezes um gerenciamento distribuído, determinando, desta forma, que a solução seja eficiente e, principalmente, escalar.

Um outro atributo bastante importante em uma plataforma que se propõe ao gerenciamento da rede está relacionado à heterogeneidade de fabricantes dos equipamentos que compõem a rede. Alinhando todos estes aspectos inerentes e desejáveis em uma solução de gerenciamento, em [CIS 06a] é apresentada uma listagem das funcionalidades que devem ser disponibilizadas em um sistema de gerenciamento padrão:

- uma ferramenta capaz de descobrir de forma automática os elementos que compõem a rede, comumente conhecido como *Discovery* da rede;
- um mapa topológico da rede mostrando, minimamente, a forma como os equipamentos estão interconectados;
- um módulo para tratamento de eventos;
- um coletor de dados de desempenho, com a possibilidade de visualização gráfica;
- Um navegador sob os dados de gerenciamento.

Neste âmbito, as soluções mais consolidadas no mercado e que merecem destaque, são:

- *Micromuse* da *Netcool*;
- plataforma *Orion*, da *SolarWinds*;
- *TNG Unicenter*, *Computer Associates* (CA);
- *Tivoli NetView*, da *International Business Machine* (IBM);

- *OpenView*, da *Hewlett-Packard* (HP).

Estes conceitos de gerenciamento são de extrema pertinência, pois provêem as diretrizes para a gestão efetiva da infra-estrutura de rede das corporações. Desta forma, enquadram-se em dois cenários distintos: grandes organizações, que dependem fortemente de sua infra-estrutura de Tecnologias da Informação e Comunicação (TIC) para operacionalizar seus processos, e as empresas provedoras de serviços de telecomunicações, que necessitam gerir sua infra-estrutura em um nível ainda mais aprofundado, pois têm como produto final à oferta de recursos de Tecnologia da Informação (TI).

No segundo contexto apresentado acima, observa-se que a necessidade de gerenciamento transcende a gestão tradicional de equipamentos e desempenho para o correto funcionamento do parque tecnológico. Efetivamente, necessita-se gerenciar em um contexto de maior amplitude, observando o comportamento da infra-estrutura em conformidade com a percepção que os clientes têm sobre o serviço de TI que contratam.

Neste sentido, surgem plataformas ainda mais especializadas, as quais têm como objetivo principal à observação da forma como os recursos estão sendo utilizados, manipulando-os dinamicamente para adequação as exigências impostas pelos usuários no momento da contratação do serviço. Tais plataformas focam, na grande maioria dos casos, na adição de arquiteturas de engenharia de tráfego, objetivando manter o dimensionamento e provisionamento dos recursos dentro dos níveis exigidos.

Obviamente, estas diferenciações na forma de perceber o gerenciamento não as desassociam dos contextos apresentados. Na verdade, são complementares. É conveniente, ainda, destacar que os apresentados por último focam no aprimoramento na forma de gerir a infra-estrutura segundo o cenário e as exigências que convivem as empresas provedoras de serviços de telecomunicações na atualidade.

## 2.2 QUALIDADE DE SERVIÇOS (QOS)

Decorrente ao contexto de gerenciamento sob a ótica do usuário final, surge um conceito bastante em voga na última década: QoS. Segundo [WAN 01], QoS pode ser definido como a capacidade de prover garantia de recursos e diferenciação de serviços em uma rede de comunicação.

Em [CIS 03], é apresentada uma definição um pouco mais ampla:

> Qualidade de Serviços refere-se à capacidade de uma rede em prover da melhor forma possível um determinado serviço, para um determinado tráfego selecionado, tendo como objetivo primordial o provisionamento de prioridade, incluindo largura de banda dedicada, atraso e *jitter* controlados e melhoramento nas características de perda.

A *International Telecommunication Union* (ITU) em sua recomendação G.1000 define QoS separadamente, como apresentado abaixo:

- qualidade é a totalidade de características de uma entidade que carrega habilidade de satisfazer determinadas necessidades declaradas e implicadas;
- qualidade de Serviço é o efeito coletivo sob o desempenho do serviço, o qual determina o grau de satisfação de um usuário em relação ao serviço.

*Räisänen* [RÄI 03] apresenta diferentes pontos de vista acerca de QoS:

- exigências de QoS do usuário ou cliente são os relatos dos níveis de qualidade requeridos pelas aplicações do usuário/cliente de um serviço, os quais não precisam ser expressos, necessariamente, de forma técnica;
- QoS oferecido ou planejado pelo provedor é um relato do nível de qualidade esperado a ser oferecido ao cliente pelo provedor de serviço;
- QoS obtido ou alcançado pelo provedor é um relato do nível atual de qualidade obtido pelo cliente;

- QoS percebido pelo usuário/cliente é um relato expressando o nível de qualidade observado pelo usuário a partir de sua experiência com o serviço a ele ofertado.

Basicamente, QoS surge como o conceito determinante no contexto das redes convergentes, impondo cada vez mais responsabilidades sobre os provedores de serviço, principalmente no que diz respeito à manutenção dos níveis de qualidade contratados pelo usuário/cliente.

## 2.3 GERENCIAMENTO DE NÍVEIS DE SERVIÇO

Segundo Morris et al. [MOR 01], *Sevice Level Management* (SLM) são procedimentos aplicados para assegurar que os níveis adequados de serviços sejam prestados a todos os usuários, levando em consideração a prioridade relativa e a importância comercial de cada um. Em grande parte dos casos, os níveis de serviço são definidos em termos da disponibilidade, capacidade de resposta, integridade e segurança oferecidas aos usuários do serviço. Tais critérios devem ser considerados e mensurados a partir dos objetivos específicos que foram delineados para a aplicação fornecida.

No contexto de controle do SLM, existem dois conceitos que são complementares. O primeiro deles trata-se de um instrumento jurídico que rege o controle do SLM de forma genérica, conhecido como o *Service Level Agreement* (SLA) ou Acordo de Níveis de Serviços, consolidado na forma de um contrato firmado entre a empresa provedora de serviços de TI e seus clientes.

Embutido no escopo de um SLA surge o segundo conceito, também conhecido como *Service Level Specification* (SLS) ou Especificação de Níveis de Serviço. O SLS caracteriza tecnicamente os parâmetros de desempenho firmados entre usuário e provedor de serviço. A seguir são apresentados os principais aspectos que caracterizam os SLA e SLS.

### 2.3.1 Acordos de Níveis de Serviço

Os acordos de níveis de serviço, SLA, são essenciais para o gerenciamento da qualidade de serviços prestados ou contratados por uma organização de TI [MOR 01]. Nesses acordos, clientes e provedores de serviço estabelecem um entendimento mútuo e definem um nível alvo de desempenho. O SLA é colocado geralmente como anexo do contrato geral entre cliente e o provedor de serviços e tem por objetivo especificar os requisitos mínimos aceitáveis para o serviço proposto. O não cumprimento do SLA implica em penalidades, estipuladas em contrato, para o provedor do serviço. A Figura 1 apresenta os diferentes atores que compõem os diferentes relacionamentos entre SLA, segundo [MAR 02].

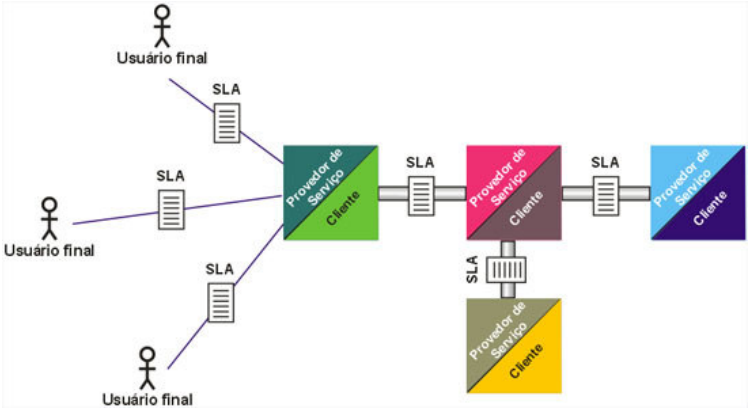



**Figura 1** Envolvimento entre clientes e provedores de Serviço.

Segundo *MORRI et. al.* [MOR 01], o SLA é a proteção contra o "fantasma da expectativa", peculiaridade da natureza humana de sempre querer mais e melhor. Especificamente nos serviços de TIC, se a disponibilidade de uma aplicação chave aumentar drasticamente, mais do que jamais solicitado antes, logo os clientes irão acostumar-se com esse nível de disponibilidade e começarão a exigir um nível de disponibilidade ainda mais elevado, podendo difamar o provedor de serviço se não obtiverem tais níveis. Os princípios do fantasma da expectativa são:

- quando as expectativas são alcançadas, elas aumentam. As pessoas nunca estão satisfeitas;
- as pessoas chateiam-se com expectativas frustradas;

- na ausência de fatos contraditórios, as expectativas se baseiam naquilo que é desejável, e não no que é possível.

Assim, o SLA surge como um instrumento poderoso que dá continuidade aos contratos nele firmados e documentados. Mais especificamente, um SLA bem redigido não define as expectativas, mas também um conjunto de indicadores aceitáveis e mutuamente acordados de qualidade de serviços.

### 2.3.2 Especificação de Níveis de Serviço

No sentido de minimizar as deficiências intrínsecas ao SLA surge o conceito de SLS, que consiste em especificações técnicas do serviço contratado. Um SLS possui várias informações, como: escopo geográfico, identificação do fluxo de dados, perfil de tráfego (taxa, rajada, etc.), tratamento de tráfego submetido em excesso, garantias de desempenho (vazão, atraso, etc.) e programação do serviço (início e duração) [KAM 01]. Um serviço deve ser definido sem ambigüidade e se possível baseado em um padrão, com uso de um SLS. Segundo [MAR 02], os seguintes tipos de informações devem ser descritos em um SLS:

- as métricas de QoS e seus respectivos limiares, os quais deverão ser garantidos pelo Provedor de Serviço;
- método de medição de desempenho do serviço, período de mensuração e fornecimento de relatórios;
- agendamento de serviços.

O SLS também define os compromissos sobre parâmetros agregados (por exemplo: tempo máximo de indisponibilidade para todos os pontos de acesso a serviço). Além disso, o SLS deve suportar diferentes modelos de interconexão de rede, assim como diferentes modelos de tráfego. Uma solução fim-a-fim para o gerenciamento de SLA requer que serviços sejam mapeados, assim como os parâmetros SLS. Dentro deste cenário, organismos de

padronização e projetos de pesquisa têm concentrado esforços. No IETF, grupos de trabalho surgiram, dos quais destacam-se:

- *Traffic Enginneering Working Group* (TEWG);
- *Realtime Traffic Flow Measurement* (RTFM);
- *IP Performance Metric* (IPPM);
- *Remote Network Monitoring* (RMONMIB).

O 3GPP, acrônimo do projeto intitulado *Third Generation Partnership Project*, apresenta a classificação de quatro classes de tráfego, baseada em *delays* requeridos, cada uma delas suportando aplicações tolerantes a erro ou aplicações intolerantes a erro [3GP 00]. A Tabela 1 apresenta a classificação do 3GPP.

**Tabela 1** Classificação 3GPP [MAR 02].

| Classe de Tráfego | Classe Conversacional | Classe *streaming* | Classe Interativa | *Background* |
|---|---|---|---|---|
| | RT Conversacional | RT *Streaming* | Melhor esforço interativo | Melhor esforço background |
| | Atraso < 150 ms | Atraso < 1 sec. | Atraso < 1 sec. | Não garantido |
| Características fundamentais | Preserva a relação de tempo (variação) entre as entidade de informação do fluxo (limitado e baixo atraso). | Preserva a relação de tempo (variação) entre as entidade de informação do fluxo. | Padrão de resposta a solicitações.<br><br>Preserva o conteúdo do *payload*. | O destino não espera pelos dados em um tempo certo.<br><br>Preserva o conteúdo do *payload*. |
| Aplicações tolerantes a erros | Voz/vídeo | Streaming de vídeo / vídeo | Mensagem de voz | Fax |
| Aplicações intolerantes a erro | telnet, jogos interativos | FTP, imagem estática, *paging* | Web browser, e-commerce, servidor de acesso a e-mail. | Chegada de e-mail, notificação. |

Outro projeto bastante relevante neste segmento é o *Traffic Engineering for Quality of service in the Internet at Large* (TEQUILA), o qual teve como meta primária o desenvolvimento de uma arquitetura integrada e técnicas associadas para prover QoS fim-a-fim em uma rede IP baseada em *diffserv* [TRI 02]. O consórcio TEQUILA é composto pela

*Alcatel*, *Algosystems* S.A., *FT-R&D*, *IMEC*, *NTUA*, *RACAL*, *UCL*, *TERENA* e *UniS*. A Tabela 2 apresenta os parâmetros de configuração do SLS propostos pelo TEQUILA.

**Tabela 2** Classificação TEQUILA [MAR 02]

| | Serviço de transferência de dados em alta velocidade | Largura de banda para serviços de dados | Serviço de garantia de taxa mínima | Seviços Olímpicos Qualitativos. | Seviços "Funil" |
|---|---|---|---|---|---|
| Comentário | Exemplo de um VLL unidirecional, com garantias qualitativas | Serviço com apenas um estrito *throughput* garantido. TC e ET não são definidos mas o operador pode definir um como proteção | Pode ser usado para um volume de tráfego FTP, ou vídeo adaptativo com um mínimo *throughput* requisitado | Em significados qualitativos são diferenciados nas seguintes aplicações: *On-line Web browsing*, Tráfego de *E-mail* | Trata-se de um serviço de proteção; restringe a quantia de tráfego que entra na rede do cliente |
| Escopo topológico | (1\|1) | (1\|1) | (1\|1) | (1\|1) ou (1\|N) | (N\|1) ou (all\|1) |
| Descritor de fluxo | EF, S-D IP-A | S-D IP-A | AF1x | MBI | AF 1.x |
| Descritor de tráfego | (b,r) e. g. r=1 | NA | (b,r) | (b,r), r indica a taxa mínima de informação comprometida | (b,r) |
| Tratamento de Excesso | Descartar | NA | Remarcação | Remarcação | Descarte |
| Parâmetros de desempenho | D = 20 (r=5, q=10e-3), L = 0 (R = r) | R = 1 | R = r | D= baixo L = baixo (*gold/ green*), D= médio L = baixo (*silver/ green*) | NA |
| MBI, diário 9:00 – 17:00 | MBI | MBI | | MBI | MBI |
| Confiança | MBI, MTD = 2 dias | MBI | MBI | MBI | MBI |

(b,r): profundidade do balde de *token* e taxa (Mbps), p: taxa de pico, D: *delay* (ms), L: probabilidade de perda; R: *throughtput* (Mbps), t: intervalo de tempo (min), q: *quantile*, S-D: origem e destino, IP-A: endereço IP; MBI: Talvez seja indicado; NA: Não aplicável; MTD: Máximo tempo *down* (por ano), ET: tratamento de excesso, TC:Conformidade de tráfego.

O consórcio *Adaptive Resource Control for QoS Using an IP-based Layered Architecture* (AQUILA), também proporcionou contribuições no sentido de padronização para representação de SLS [AQU 00]. Tais contribuições foram submetidas ao IETF sob

forma de *draft*. Porém, após seis meses da submissão, a *draft* expirou. O principal objetivo desta *draft* era descrever o uso de SLS no *framework* proposto pelo consórcio AQUILA, aliando a este trabalho um *feedback* a *draft* "*Service Level Specification Semantics and Parameters*" proposta pelo TEQUILA.

## 2.4 ENGENHARIA DE TRÁFEGO EM REDES IP

A partir do exposto nas seções anteriores, fica explícita a preocupação por parte dos provedores de serviço com a qualidade dos serviços prestados sob a perspectiva do usuário final. Desta forma, apresentou-se inicialmente o conceito macro de "qualidade" aplicado a este contexto e os instrumentos utilizados para formalizar, em diferentes níveis, o desejo de ambas as partes (cliente e provedor) com relação aos serviços contratados.

Sob esta ótica, começam a surgir iniciativas no âmbito científico-acadêmico com o objetivo de propor uma nova aproximação para o adequado condicionamento de recursos. Neste ponto, identifica-se como principal alternativa à adaptação dos conceitos e técnicas pertinentes à engenharia de tráfego, aplicados ao gerenciamento de redes IP. Basicamente, a engenharia de tráfego é determinada como um conjunto de técnicas e ferramentas que podem ser utilizadas para otimizar o desempenho de uma rede IP operacional.

O *overview* sobre engenharia de tráfego [AWD 02] apresentado sob a forma de RFC, do grupo de trabalho TEWG do IETF norteia as principais recomendações no contexto de aplicação das técnicas relacionadas à engenharia de tráfego em redes IP. Esta RFC define engenharia de tráfego como:

> A aplicação da tecnologia e de princípios científicos para mensurar, caracterizar, modelar e controlar o tráfego Internet.

De acordo com [WAN 01], o objetivo principal da engenharia de tráfego está centrado em reduzir o congestionamento e aprimorar a utilização dos recursos da rede, mantendo um gerenciamento atento em relação à distribuição do tráfego. Desta forma, a engenharia de

tráfego é observada como subsídio para identificação e estruturação dos objetivos e prioridades que devem ser aprimorados sob a perspectiva da experiência do usuário final.

Em [RÄI 03] é apresentada uma reformulação das definições previamente apresentadas, caracterizando engenharia de tráfego como meios sistemáticos de análise do estado da rede, extração de conclusões a partir das análises efetuadas e a materialização de reconfiguração da rede. O autor identifica ainda algumas razões para esta abordagem:

- aumento no volume de tráfego local, com a adição de novos clientes nos *Point of Presence* (PoP), ou Pontos de Presença;
- alteração na situação dos recursos, como a alteração nos provedores da capacidade das linhas concedidas;
- alteração na distribuição do tráfego agregado (localmente ou globalmente), como, por exemplo, a adoção a novos tipos de serviço;
- condições excepcionais como o mal funcionamento de um dispositivo (por exemplo, um roteador).

Outras áreas de concentração da engenharia de tráfego estão centradas no melhoramento e manutenção na confiabilidade das operações da rede em casos de situações excepcionais como, por exemplo, a resultante da falha de algum equipamento. Nestes casos é requerida a habilidade de detecção da falha, assim como a habilidade de atuar baseado nesta informação.

### 2.4.1 O processo de engenharia de tráfego

O modelo do processo de engenharia de tráfego é iterativo [AWD 99], estando ilustrado na Figura 2 . Segundo [AWD 02], o processo de engenharia de tráfego é definido e constituído pelos seguintes componentes:

- definição de políticas de controle relevantes: Este pode ser considerado como o ente externo do atual processo de engenharia de tráfego, responsável por controlar

seu progresso. Naturalmente, políticas de controle devem ser e são ajustadas baseadas no desempenho observado a partir da rede que está sendo controlada;

- mecanismos de *feedback*: mecanismos responsáveis pela aquisição de dados de desempenho da rede de produção;
- análise do estado da rede: caracteriza-se pela análise e caracterização da carga de trabalho do tráfego passante;
- otimização de desempenho: partindo das etapas anteriores do processo, caracteriza-se por materializar as configurações necessárias, no sentido de otimizar o desempenho da rede.

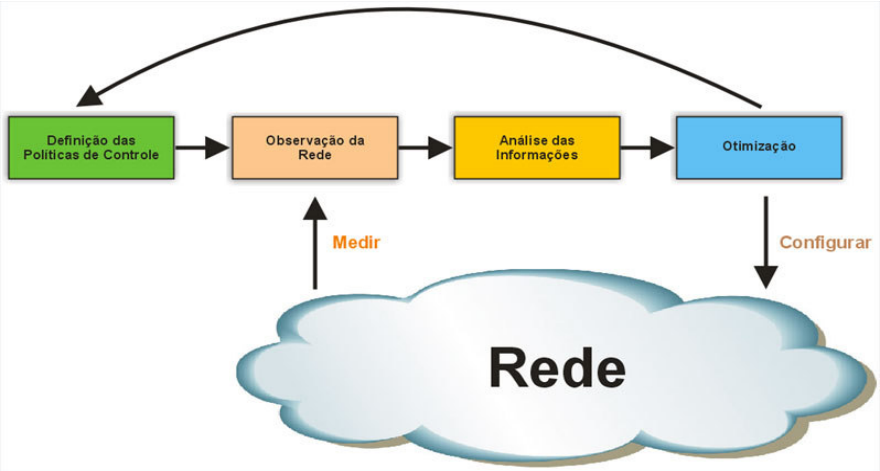



**Figura 2** O processo de engenharia de tráfego [AWD 02]

O componente de definição das políticas de controle depende de muitos fatores incluindo o modelo de negócio, o custo de estrutura da rede, as restrições de operação, o modelo de utilidade e os critérios de otimização.

No segundo componente é destacado o fato de que, caso não estejam disponíveis dados de desempenho coletados da rede, devem ser utilizadas cargas de trabalho sintéticas a partir de cargas de trabalho previamente esperadas [AWD 02]. As cargas de trabalho sintéticas podem ser derivadas pela estimação ou extrapolação, usando dados empíricos prévios. A derivação de tais dados pode ser obtida utilizando modelos matemáticos das características do tráfego.

Para o componente de análise, pode-se utilizar um modelo proativo ou reativo. No modo reativo, a análise está centrada em identificar possíveis locais da rede que apresentem um desempenho sub-ótimo, traçar qual é a causa raiz e testar diferentes caminhos para solucionar o problema [RÄI 03]. No modelo proativo, o objetivo é identificar alvos de otimização, no sentido de antever e prevenir futuros problemas de desempenho.

Por último, o componente de otimização inclui os meios para seleção do método real a ser usado para otimização de desempenho da rede. Nesta fase, o operador de rede pode ter um grande benefício utilizando o modo de análise proativo, se os meios de análise são suficientemente bons. Em [RÄI 03] é destacado que, de forma otimista, a análise de dispositivos pode ser usada como meio de comparação para antecipar configurações potenciais da rede.

## 2.5 OBTENÇÃO DE INDICADORES DE DESEMPENHO

A obtenção de dados de desempenho pode ser feita a partir de múltiplas fontes e em diferentes níveis de abstração [RÄI 03]. Sob a perspectiva de um método mais detalhado, os elementos de rede podem ser individualmente consultados para aquisição de informações. Da mesma forma, para a maioria das características, a obtenção de dados de desempenho em um nível mais alto de abstração, pode ser obtida com o uso de sistemas *Network Management System* (NMS), ou Sistema de Gerenciamento de Rede.

Os sistemas de gerenciamento podem prover tipicamente médias e análises de tendência a partir de um *status* abstrato e momentâneo da rede. No caso das redes multi-serviços, o desempenho da rede pode ser monitorado em nível do tráfego agregado para um determinado domínio autoritativo e finalmente, esta observação de comportamento pode ser formulada em termos da qualidade do serviço que está sendo prestado.

Tipicamente, os elementos de rede (um roteador, por exemplo) provêem informações de gerenciamento através do protocolo *Simple Network Management Protocol* (SNMP), acessando os dados contidos nas *Management Information Base* (MIB), ou Base de

informações de Gerenciamento. Porém, devido a grande diversidade de equipamentos disponíveis, torna-se comum encontrar interfaces não padronizadas, que utilizam tecnologias proprietárias, adicionando um alto grau de dificuldade na aquisição de informações. A Tabela 3 , apresenta uma caracterização dos diferentes níveis de monitoramento possíveis.

**Tabela 3** Caracterização dos níveis de monitoramento da rede [RÄI 03].

| Nível de monitoramento | Características típicas que são mensuradas |
|---|---|
| Para elemento de rede | Carga completa e estatística para cada tráfego agregado. |
| Sistemas de Gerenciamento IP | Dados da rede como um todo, médias e análises de tendências. |
| Desempenho agregado | Atraso, *jitter*, perda de pacotes e largura de banda disponível em um domínio de rede. |
| Nível de serviço | Especificação de Níveis de Serviço |

Especificamente, o nível agregado de desempenho está intimamente relacionado ao desempenho da rede para suporte à qualidade de serviços agregados. Geralmente, existem três diferentes métodos, os quais podem ser usados para estimativa do nível de desempenho agregado, são eles:

- medição passiva (não intrusiva);
- medição ativa (intrusiva);
- medição *PiggyBacking*.

Os dados advindos da medição obtidos por um ou mais dos mecanismos acima citados, podem ser processados com diferentes propósitos no sentido de obtenção de características relevantes.

### 2.5.1 Medição passiva

A premissa básica da técnica de medição passiva está centrada no fato de que esta não requer a injeção de tráfego adicional para mensuração da rede operacional. Desta forma, o

tráfego habitual em uma rede operacional pode ser monitorado, podendo ser utilizados um ou mais pontos de medição em um determinado domínio de rede.

Segundo [COS 05], as medições passivas são ideais para monitoramento do tipo de tráfego. Porém, nem sempre todos os pontos da rede dispõem da tecnologia necessária, ou seja, muitas vezes torna-se necessária uma infra-estrutura considerável de *hardware* e *software* específica para tal atividade. Nesta área, destacam-se os trabalhos desenvolvidos pelo projeto *Passive Measurement and Analysis* (PMA) [MCG 00] do *National Laboratory for Applied Network Research* (NLANR).

No escopo deste tipo de medição, deve ser considerado o número de pontos de medição, sendo estes determinantes no tipo de informação de desempenho que pode ser extraída da rede. Com um único ponto de medição é possível o monitoramento das características do tráfego passante e estatística do protocolo [RÄI 03]. Como exemplo de monitoramento de estatística com o uso do *Real Time Protocol* (RTP), destaca-se a medição de variação no atraso e perda de pacotes. O presente trabalho trata da caracterização de informações sobre o tráfego passante, especificamente a medição de tráfego baseada em fluxos (com foco em grande fluxos), e será melhor detalhada posteriormente.

Para o caso de se utilizar dois pontos de medição, torna-se possível à mensuração de atraso em uma via, sem a necessidade adicional de colunas de *timestamp*. Para isso, torna-se necessário prover um mecanismo de sincronização entre os pontos de medição. Tal sincronismo dá-se em dois contextos. O primeiro deles devido à necessidade de sincronismo temporal permanente entre as entidades de medição, para desta forma obter-se precisão na medição. O segundo ponto de simultaneidade entre as entidades de medição, consolida-se na necessidade de marcação dos pacotes que estão sendo considerados na medição. Em [NIC 04], é apresentada uma proposta neste sentido.

Embora a medição passiva não adicione tráfego à rede, esta técnica requer maior processamento que a técnica de medição ativa, pois, principalmente em redes de núcleo com alta capacidade, necessita manipular uma quantidade muito grande de dados. Por este motivo, deve-se ter um cuidado adicional com os pontos de medição, levando em consideração na implementação, mecanismos pertinentes de alto-desempenho e a aplicação de técnicas

fundamentadas em modelos amostrais. Por estes fatores, a medição passiva foi um importante alvo de padronização na última década, consolidando as RFC's 1757, 2021 e 3432, no IETF.

### 2.5.2 Medição ativa

Ao contrário da técnica de medição passiva, a medição ativa, propõe-se a obter indicadores de desempenho, a partir da transmissão controlada de tráfegos de teste através da rede que se deseja analisar [RÄI 03]. Segundo [CAL 05], os dados obtidos a partir da medição ativa provêem as seguintes informações relacionadas à rede: topologia, largura de banda disponível e no gargalo, atraso em uma via, atraso de ida e volta, perda, variação no atraso (*jitter*) e grau de desordem dos pacotes.

Tradicionalmente, o tráfego gerado pelo o processo de medição ativa é formado por pacotes denominados de *probes* (sondas). Tais *probes* devem ser dimensionados e injetados na rede seguindo critérios bastante rígidos, para que a medição possa aproximar-se da forma mais precisa possível de um tráfego real. Obviamente, tal preocupação está intimamente relacionada aos aspectos menos favoráveis desta técnica: o consumo intrínseco de recurso da rede e a inserção de competitividade com o tráfego corrente.

Um dos problemas clássicos nas medições ativas, especificamente no caso das métricas unidirecionais relacionadas a tempo (atraso em uma via, por exemplo), é a necessidade da sincronização temporal dos pontos envolvidos no processo de medição. Como contorno para este empecilho é necessário um alto investimento para aquisição de *Global Positioning System* (GPS) ou outras fontes confiáveis de tempo. Destacam-se nesta área a iniciativa do *Active Measurement Project* (AMP) [MCG 00], *Surveyor* [KAL 99] e *TTM-RIPE* [GEO 01].

Como aproximações mais conhecidas da aplicação de medições ativa e comumente implementadas nos sistemas operacionais atuais, destacam-se os utilitários *Ping* e *Traceroute*, os quais utilizam pacotes *Internet Control Message Protocol* (ICMP) para determinar atraso de ida e volta e topologia de rota da rede. Recentemente, foram desenvolvidas ferramenta de medição ativa [MAN 04][MAO 03] para emular o tráfego de aplicações específicas, com o

objetivo de, a partir dos resultados obtidos, determinar o grau de confiabilidade sob a perspectiva de aplicação. Outras ferramentas como [DOW 99][DOV 04], utilizam esta técnica para mensurar a largura de banda disponível e a largura de banda no gargalo. No contexto de medir atraso em uma via, destaca-se a arquitetura apresentada em [SHA 04]. A *Universal Active Measurement Architecture* (UAMA) [SAN 07b] destaca-se como um sistema capaz de integrar diferentes métricas de desempenho em uma única plataforma, apresentando características de expansibilidade e, principalmente, mantendo conformidade com as recomendações do IETF.

Esta técnica tornou-se bastante atrativa, principalmente para o monitoramento de desempenho das aplicações emergentes (VoIP, *vídeo streaming*, entre outras), que necessitam de uma averiguação cada vez mais condizente com a perspectiva de seus usuários finais. Neste contexto, tornou-se alvo de padronização, possuindo um grupo de trabalho específico no IETF, intitulado de IPPM [PAX 98].

### 2.5.3 *Piggybacking*

A técnica *Piggybacking* para medição é considerada, no momento, uma abordagem mais orientada à pesquisa, onde informações de *timestamps* e números de seqüência são anexados ao *payload* dos pacotes [JOR 00]. O conceito de *piggybacking* atrelado ao *Transmission Control Protocol* (TCP) tem relação à confirmação de recebimento de pacotes, especificamente, um pacote que trafega em um sentido, confirma o último pacote recebido no sentido contrário.

Segundo [RÄI 03], nesta técnica de medição, devem ser consideradas as sobrecargas adicionadas ao processo de medição propriamente dito. Neste sentido, deve-se observar atentamente ao processamento extra requerido, tanto para inserir, como para remover as informações, anexadas ao *payload* dos pacotes, são necessárias para efetuar-se a medição. O consumo extra de processamento deve ser contabilizado e devidamente considerado na extração das informações de medição.

### 2.5.4 Comparação entre as técnicas de medição

Como forma de tornar mais visível as diferenças existentes entre as três diferentes técnicas de medição apresentadas buscou-se traçar um comparativo no sentido de situar onde cada uma das técnicas melhor se enquadra para a obtenção dos principais indicadores de desempenho.

Inicialmente, é importante enfatizar quais são os principais indicadores de desempenho (métricas) considerados na obtenção de informações de uma rede em análise. Segundo [ITU 02], na perspectiva de provisionamento de uma rede IP para estabelecimento de níveis adequados de QoS, devem ser consideradas as seguintes métricas de desempenho: atraso fim-a-fim para a transferência de um pacote IP, variação no atraso, taxa de perda de pacotes IP e taxa de erro em pacotes IP.

Como já citado, o grupo de trabalho IPPM também tem mantido foco na padronização de tais indicadores de desempenho. Neste sentido, cabe salientar, além dos apresentados acima, esforços na definição de métodos eficientes para mensuração de métricas relacionadas à largura de banda, mais precisamente, na obtenção da largura de banda disponível (*throughput*). Também se tornou objeto de padronização a métrica relacionada ao grau de desordem dos pacotes.

Fazendo uma abordagem própria, [RÄI 03] elabora um comparativo entre os três diferentes modos de mensurar a rede sob a perspectiva de determinar qual a aplicabilidade de cada uma na quantificação de quatro diferentes métricas: atraso em uma via, variação no atraso, perda de pacotes e *throughput*. A Tabela 4 apresenta este comparativo.

Obviamente, nenhuma das técnicas apresenta-se como uma solução completa. Este fato caracteriza-se pela natureza e particularidades das métricas. Além disso, deve ser considerada a amplitude do processo de mensuração e o seu objetivo. Na prática, a utilização do método de medição ativa, por exemplo, torna eficiente a obtenção de indicadores de desempenho em um contexto fim-a-fim, principalmente em cenários intra-domínios. Também é considerada a melhor aproximação para a percepção de desempenho sob a ótica de usuários finais.

**Tabela 4** Comparação entre as técnicas de medição [RÄI 03]

| Técnica | Atraso em uma via | Variação no atraso (*jitter*) | Perda de pacotes | *Throughput* |
|---|---|---|---|---|
| Medição Passiva | (1) | (2) | (2) | X |
| Medição Ativa | X | X | X | (3) |
| Piggybacking | X | (3) | X | (3) |

(1) Necessita de dois pontos de medição para prover resultados confiáveis.
(2) É possível com apenas um ponto de medição, utilizando-se *timestamps* e números de seqüência.
(3) Indiretamente possível.

Por outro lado, o método de medição passiva tem uma amplitude mais reduzida, sendo bastante restritivo na obtenção de indicadores que necessitem de mais de um ponto de medição. Porém, é de grande aplicabilidade em redes de núcleo onde se deseja observar o tráfego em trânsito, caracterizando seu comportamento para os mais diversos fins (contabilização do volume de dados, caracterização do tráfego, contabilização, proteção no caso de auditorias por quebra de SLA, entre outros).

Resumidamente, os métodos possuem particularidades que denotam vantagens e desvantagens inerentes. Porém, deve ficar enfatizado que as técnicas são complementares, ou seja, são diferentes meios para se atingir um mesmo objetivo. Assim, é considerada como boa prática à utilização em conjunto destas técnicas, pois permite a extração de informações complementares da rede, obtendo-se diretamente uma visão mais ampla em termos de confiabilidade e completitude.

## 2.6 MEDIÇÃO DE TRÁFEGO BASEADA EM FLUXOS

Partindo do apresentado neste capítulo, pode-se inferir que existem diversos esforços no sentido de obter-se um diagnóstico com alto grau de precisão das condições da rede. Também é notória a inexistência de uma solução unificada, ou seja, para consolidação do processo de medição que conduza a resultados satisfatórios, é imprescindível à combinação de diferentes métodos e técnicas.

Neste contexto, a medição de tráfego baseada em fluxos tem sido um dos principais objetos de interesse dos operadores de rede, considerado como um dos caminhos mais pertinentes para entendimento do comportamento dos dados que trafegam pela rede [BRO 01]. As informações do tráfego são consideradas vitais para observação de problemas (detecção de comportamentos não usuais), para coleta de dados de uso (detecção de incidentes de segurança e bilhetagem) e planejamento de capacidade.

Em [BRO 01] são destacados três caminhos para mensurar o tráfego de uma rede. A primeira aproximação está relacionada à escrita de cópias integrais dos pacotes ou de seus cabeçalhos em arquivos de coleta, utilizando ferramentas como o *tcpdump* [JAC 89]. Uma vez que as informações foram coletadas, estas podem ser analisadas posteriormente, buscando, geralmente, traçar correlações entre diferentes pontos de medição.

Esta abordagem possui duas vantagens diretas. A primeira delas relaciona-se ao fato de ter-se a possibilidade de uma análise mais aprofundada sobre o comportamento do tráfego, uma vez que o conteúdo dos pacotes pode ser guardado em sua totalidade. A segunda vantagem a ser destacada, está relacionada à possibilidade dos pacotes serem capturados em *links* de alta capacidade, uma vez que o fator limitador está relacionado à capacidade de escrita em disco. Porém, a progressão de crescimento dos arquivos de coleta é via de regra inviável pelo fato de armazenar cada pacote, individualmente. Além disso, a análise deve ser efetuada de forma particular em cada interface de rede, impossibilitando a observação de um comportamento fim-a-fim.

Uma segunda abordagem para medição do tráfego da rede é padronizada pelo IETF e intitulada como *Remote Network Monitoring* (RMON) (RFC 2819). Um agente RMON, que usualmente é adicionado nos *switches* e/ou roteadores, implementa a MIB RMON [WAL 00], permitindo ao gerente da rede determinar níveis de tráfego em segmentos particulares, carga total de tráfego de entrada e saída para *hosts* ocupados, carga de tráfego entre um par de *hosts* para diferentes protocolos, entre outras possibilidades. Porém, a RMON não consegue prover nenhum tipo de potencialidade para mensurar fluxos propriamente ditos.

A terceira aproximação para medição de tráfego, que será foco do presente trabalho, refere-se ao armazenamento dos fluxos observados em um determinado segmento de rede. Segundo [BRO 01], a noção de fluxos de tráfego tem sido utilizada há muitos anos, porém

fluxos podem ser definidos sob diferentes horizontes. Em [HAN 99], fluxos são vistos como um grupo de pacotes entre duas entidades (origem/destino), definidos por seus atributos de origem e destino e seu período de início e fim, sendo bi-direcionais, ou seja, observados nas duas direções.

No contexto de roteamento, um fluxo é caracterizado como uma seqüência de pacotes direcionados de um IP de origem a um outro de destino, sendo, desta forma, unidirecional. Atualmente, é recomendada a identificação de fluxos em uma granularidade mais fina, onde os fluxos são identificados por uma tupla de cinco atributos: protocolo, endereço de origem e destino, número de porta de origem e destino. Neste sentido, a medição baseada em fluxos trata basicamente na observação de pacotes no tráfego passante, na construção de tabelas de informações de fluxo que armazenam a combinação de endereços na *tupla* de cinco atributos, na contabilização de pacotes e volume de *bytes.*

A medição de tráfego baseada em fluxos tem se consolidado como uma técnica de grande apelo, pois pela análise dos fluxos em circulação em um segmento de rede é possível a quantificação e a qualificação do tráfego sem a necessidade da análise individual de cada interface de rede do segmento [VIE 04]. Além disso, em engenharia de tráfego, o processo de análise de fluxo tem se tornado um elemento cada vez mais importante, pois possibilita contabilizar o grau de utilização da rede, subsidiando tomadas de decisão no provisionamento de recursos.

Deve ser destacado, ainda, que no planejamento de infra-estrutura das operadoras de serviços de rede (que em grande parte dos casos é efetuado de forma empírica, sem fundamentação com base na efetiva utilização dos recursos), os resultados de medição do tráfego baseado em fluxos podem servir como um instrumento de grande relevância. A partir deles, têm-se conhecimento da real necessidade de *upgrade* da infra-estrutura, tornando possível a elaboração de um dimensionamento mais preciso, evitando custos desnecessários.

## 2.7 PADRÕES RELACIONADOS À MEDIÇÃO DE TRÁFEGO BASEADO EM FLUXOS

A presente seção tem por objetivo abordar os principais padrões propostos na comunidade científica para a medição de tráfego baseado em fluxos. Desta forma, um breve relato de cada uma das abordagens é apresentado, dando-se ênfase a suas peculiaridades.

### 2.7.1 *Real-Time Flow Measurement*

Em 1995, foi instanciado no IETF o grupo de trabalho intitulado RTFM, que tinha por objetivo a produção de uma arquitetura para medição de tráfego baseado em fluxos. A principal diretriz desta proposta aponta para o processo de medida de uso, ou contabilização, do tráfego da rede.

Seguindo as características descritas na arquitetura de medição de fluxo apresentada na RFC-2722 de outubro de 1999, existem quatro componentes básicos no processo de medição [BRO 99a]:

- gerente de medição de tráfego: é uma aplicação que tem como função orquestrar a configuração das entidades ‘Medidores’ e controlar a entidade ‘Leitor de Medidas’. Envia comandos de configuração para os medidores e supervisiona a operação de cada medidor e do leitor de medidas. É considerado como conveniente combinar as funções do leitor de medidas e do gerente em uma única entidade de rede;
- medidores: são colocados em pontos de medição estratégicos determinados, em geral, pelo operador da rede. Cada medidor armazena seletivamente as atividades da rede de acordo com a configuração previamente estabelecida. Os resultados processados e armazenados são chamados de “dados de uso”;

- leitor de medidas: é responsável por encaminhar as informações coletadas pelos medidores, tornando-as disponíveis para as aplicações de análise;
- aplicações de análise: são responsáveis por processar as informações advindas dos medidores, no sentido de prover informações e relatórios que serão utilizados como embasamento para a engenharia de rede e propostas de gerenciamento. Abaixo, são citados alguns exemplos de aplicações de análise:
    - matrizes de fluxos de tráfego: apresenta a taxa total de fluxos para a maioria dos caminhos possíveis na rede;
    - distribuição do fluxo de tráfego: sumarização das taxas de fluxos sob um determinado período de tempo;
    - dados utilizados: apresenta o volume de tráfego total enviado e recebido para um *host* particular.

Em [VIE 04] é destacado que o gerente (também referenciado por coletor) pode recuperar fluxo de dados de vários medidores e, cada medidor, pode ter seus dados recuperados por vários gerentes. O fluxo de tráfego de interesse é definido pelo usuário na forma de regras (*Rule Sets*). O relacionamento entre os elementos do processo de medição do tráfego de rede é apresentado na Figura 3 .

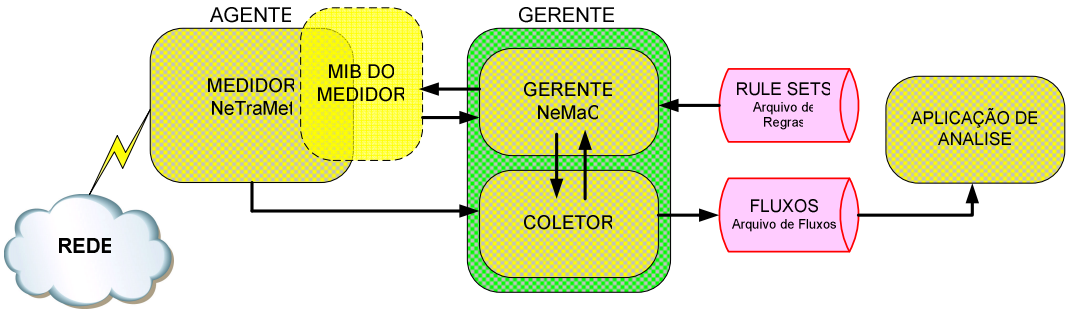



**Figura 3** O relacionamento entre os elementos do processo de medição do tráfego de rede baseado em fluxos [VIE 04]

Em outubro de 1999, o grupo de trabalho RTFM publicou a RFC 2720. A RFC 2720 descreve e define uma MIB para o controle dos medidores de tráfego, em particular para especificar os fluxos a serem mensurados [BRO 97b]. Desta forma, obtém-se diretamente um

mecanismo eficiente para coleta das informações dos fluxos a partir de um medidor, utilizando o protocolo SNMP.

### 2.7.2 *IP Flow Information Export* (IPFIX)

O grupo de trabalho IPFIX, do IETF, surge como uma proposta bastante atual de padronização no formato de exportação de informações de fluxos IP. Atualmente, o grupo de trabalho possui duas RFC´s: *Requirements for IP Flow Information Export* - RFC 3917 e *Evaluation of Candidate Protocols for IP Flow Information Export* - RFC 3955, ambas publicadas em outubro de 2004.

Segundo os proponentes desta padronização [PLO 06], existe um número significativo de sistemas que visam à exportação das estatísticas de fluxos, porém estes se diferenciam significativamente, mesmo que alguns adotem mecanismos comuns para o transporte da informação. Tais diferenças dificultam o desenvolvimento de ferramentas genéricas de análise de fluxo. Além disso, existe uma necessidade concreta da indústria de dispositivos de rede (roteadores, por exemplo), assim como na comunidade de pesquisa, por um formato padrão no transporte das estatísticas de fluxos para sistemas externos.

Sob esta perspectiva, um sistema de exportação de informações de fluxos IP inclui um modelo de dados para representação das informações de fluxo e um protocolo para transporte de tais informações. Um "exportador" potencial de informações é tipicamente um roteador ou um dispositivo dedicado a mensurar tráfego IP. Neste sentido, as informações sobre fluxo reportadas, segundo [PLO 06], devem conter:

- atributos derivados do cabeçalho dos pacotes IP, como endereço de origem e destino, protocolo e número de portas;
- os atributos geralmente conhecidos apenas pela entidade "exportadora", como portas de ingresso e egresso, máscara de rede e subrede, números de sistemas autônomos e, talvez, informações secundárias sob a camada IP.

Neste contexto, o grupo de trabalho possui alguns objetivos específicos, relacionados à padronização que propõem. É interessante demarcar tais objetivos, pois a partir destes torna-se claro o escopo de atuação do IPFIX. São eles:

- definir a noção de um padrão para fluxos IP. A definição de fluxo é, de forma prática, similar a correntemente utilizada em protocolos de exportação de informações de fluxos não-padronizados que tentaram atingir objetivos similares, porém não os documentaram;
- planejar os *encondings* de dados para suportar fluxos IPv4 e IPv6 *unicast* e *multicast*, os quais transpassam um determinado elemento de rede no nível de cabeçalhos de pacote ou em outros níveis de agregação, configuradas pelo administrador da rede;
- considerar a noção de exportação de estatísticas de fluxos baseada em amostragem de pacotes;
- identificar e endereçar os mecanismos de segurança, para que os dados dos fluxos não sejam afetados. Especificamente, determinar a tecnologia de segurança a ser utilizada na exportação da informação;
- especificar o mapeamento de transporte para carregamento das informações de fluxo;
- assegurar que o sistema de exportação seja confiável o suficiente, a ponto de que, no caso de perdas, estas sejam perfeitamente identificadas e reportadas.

Em resumo, o escopo dos trabalhos desenvolvidos pelo IPFIX se limita a identificação dos fluxos e na transferência de dados para uma entidade coletora que seja capaz de interpretar tais informações, conforme expressado na Figura 4 .

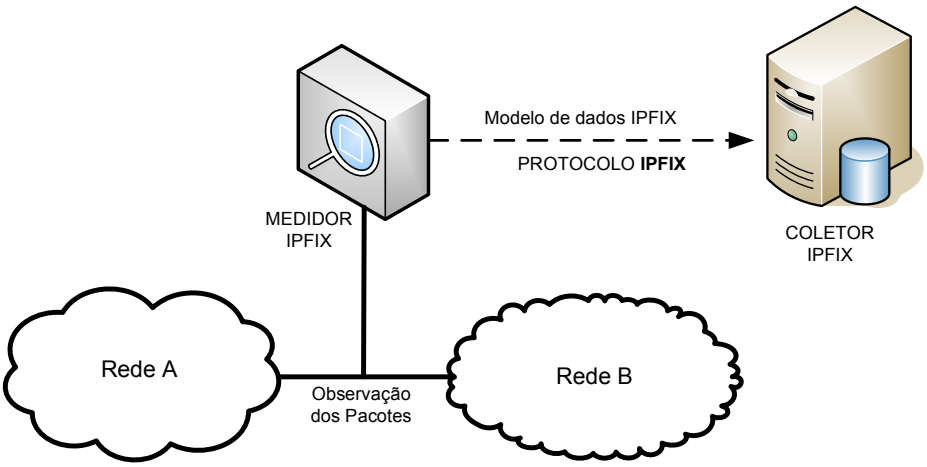



**Figura 4** Escopo de atuação do IPFIX.

## 2.8 FERRAMENTAS E TECNOLOGIAS PARA MONITORAMENTO DE TRÁFEGO BASEADO EM FLUXOS

Existem muitas implementações realizadas no sentido de prover o monitoramento de tráfego baseado em fluxos. Tais trabalhos foram desenvolvidos de forma adjacente ao surgimento de padrões nesta área. No sentido de contextualizar o trabalho, serão apresentadas, de forma sucinta, as principais ferramentas (consagradas) para o apoio ao gerenciamento de redes no contexto de monitoramento de tráfego baseado em fluxos.

### 2.8.1 *NetFlow*

O *NetFlow* é uma implementação proprietária da Cisco [CIS 06b] que tem o objetivo de prover estatísticas a respeito do fluxo de pacotes que trafegam pelos roteadores da rede. É capaz de identificar fluxo de pacotes, tanto para os pacotes IP de ingresso como para os pacotes de egresso, gerenciando a captura e exportação das informações coletadas de forma independente em cada dispositivo de rede.

Para o *NetFlow*, um fluxo da rede é definido como um *stream* unidirecional de pacotes entre uma determinada origem, para um determinado destino. A origem e o destino são definidos pelo endereço IP na camada de rede e, na camada de transporte, pelos números de porta (também para origem e destino). Especificamente, um fluxo é definido pela combinação dos seguintes campos, denominados campos-chave:

- endereço IP da origem;
- endereço IP do destino;
- número da porta na origem;
- número da porta no destino;
- tipo de protocolo na camada 3;
- *Type of Service* (ToS), ou Tipo de Serviço;
- entrada na interface lógica.

Estes sete campos-chave definem um fluxo único. Caso um determinado pacote tenha algum campo-chave que difira de um outro pacote qualquer, ambos são considerados como pertencentes a fluxos distintos.

Existem dois componentes chave no *NetFlow*: *NetFlow Cache*, responsável pelo armazenamento das informações dos fluxos IP e o *NetFlow Export* ou mecanismo de transporte para envio dos dados coletados ao coletor de gerenciamento da rede, através da *engine* de coleta. Para cada fluxo ativo uma nova entrada é armazenada na *cache*, sendo que cada uma dessas entradas possui as informações pertinentes para exportação das informações coletadas. Segundo documentação da CISCO [CIS 06b], a versão 9 do formato de exportação do *NetFlow*, serviu de base para elaboração do IPFIX.

Um outro ponto a ser destacado com relação ao *NetFlow*, são as possibilidades na forma como as informações serão obtidas a partir do tráfego passante. Basicamente, são implementadas duas abordagens para observação do tráfego. A primeira delas é determinada pela amostragem de pacotes aplicável na engenharia de tráfego ou no planejamento de

capacidade. A segunda aproximação está relacionada à filtragem seletiva, onde são determinadas as classes de tráfego que se deseja monitorar.

Efetivamente, o *NetFlow* consolida-se como uma arquitetura de extrema robustez, principalmente por suas potencialidade e por advir de uma das maiores fabricantes mundiais na área. Por outro lado, por tratar-se de uma tecnologia proprietária, exige a utilização exclusiva de equipamentos CISCO. Desta forma, torna-se proibitiva para grande parte dos cenários de rede, pois exige um alto custo para aquisição dos equipamentos. Além disso, em [VIE 04], é enfatizado que na prática apenas roteadores da série 6000 e, preferencialmente, da série 12000 comportam-se de forma adequada, ratificando ainda mais a inviabilidade financeira para grande parte dos casos.

### 2.8.2 *sFlow*

O *sFlow* é uma tecnologia para monitoramento de tráfego em redes de dados baseadas em *switches* e roteadores [PHA 01]. O sistema de monitoramento do *sFlow* consiste de um agente (embarcado em um *switch*/roteador ou em um módulo isolado) e um coletor central (também chamado de analisador *sFlow*), conforme a Figura 5 . A arquitetura e a técnica de amostragem usadas no sistema de monitoramento *sFlow* foram projetadas para prover monitoramento de tráfego em redes de alta-velocidade.

Os objetivo principais do *sFlow* abordam, especificamente, temas relacionados com:

- monitoramento do tráfego de redes com precisão para velocidades *Gigabits* ou maiores;
- escalonamento para monitorar milhares de agentes de um único coletor *sFlow*;
- baixo custo de implementação.

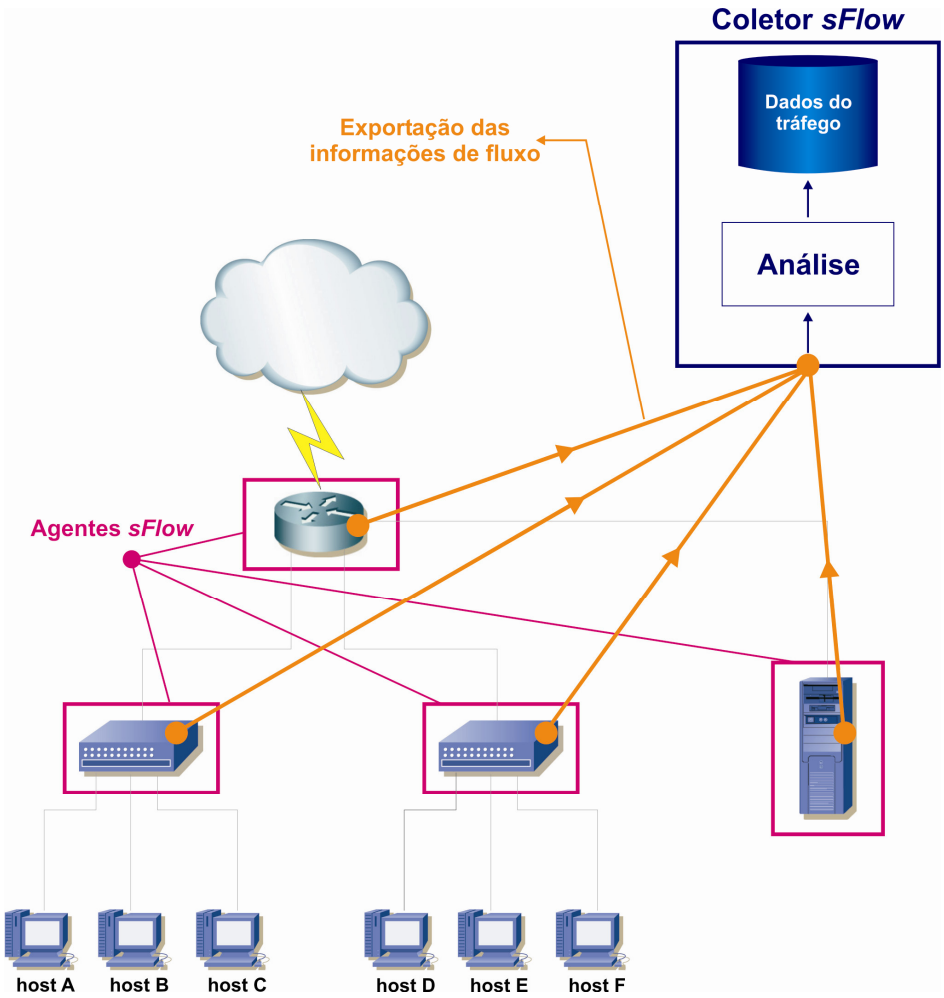



**Figura 5** Arquitetura *sFow*.

Basicamente, o agente *sFlow* usa amostragem para capturar as estatísticas de tráfego de um dispositivo monitorado. Os datagramas do *sFlow* são utilizados então para encaminhar, continuamente, as estatísticas do tráfego amostrado a um coletor para que sejam efetuadas as análises. Desta forma, consegue-se obter uma visão em tempo-real do tráfego de fluxos na rede.

Em termos de aplicabilidade, o *sFlow* se apresenta como uma alternativa de baixo custo de implementação e como um padrão, de fato, para agregação à *switches* e roteadores dos mais diversos fabricantes da área. Porém, ainda não se pode considerar como uma alternativa final, pois a sua presença nos dispositivos está longe de estar capilarizada de forma consistente a ponto de se desenvolverem aplicações voltadas especificamente a esta tecnologia.

### 2.8.3 *nTop*

O *nTop* é uma aplicação *open-source* escrita na linguagem *C*, de distribuição livre sob licença pública. Segundo [DER 00], os objetivos principais do projeto *nTop*, são:

- portabilidade entre sistemas baseados em *UNIX* e não-*UNIX*, isto é, plataforma *Windows*;
- aplicação simples e eficiente com baixo custo no uso de recursos;
- habilidade para monitorar e gerenciar a rede a partir de um local remoto sem necessidade de rodar aplicações cliente específicas para análise das informações de tráfego;
- exigências mínimas, mas com a capacidade de explorar funcionalidades da plataforma onde está alocado, como, por exemplo: *threads*;
- capacidade de apresentar os dados em um terminal baseado em caracteres ou em um navegador *Web*;
- extensibilidade com a possibilidade de agregação de *plug-in´s*.

Basicamente, o *nTop* é uma sonda que atua no tráfego da rede e monitora sua taxa efetiva de utilização. Fazendo uma analogia, o *nTop* se comporta de forma muito semelhante ao comando *TOP*[1], disponível em plataformas baseada em *UNIX*. Com interface *Web-based*, o *nTop* foca em quatro características primárias:

- medição de tráfego;
- monitoramento de tráfego;
- otimização e planejamento da rede;
- detecção de violação da segurança da rede.

---

[1] O programa *TOP* é nativo em praticamente todas as distribuições baseadas em *UNIX* e provê uma visão em tempo-real das tarefas executadas pelo sistema operacional.

O *nTop* efetua o rastreamento da rede gerando uma série de estatísticas para cada um dos *hosts* que compreende a rede em análise. A informação necessária é coletada pelo *host* rodando o *nTop* simplesmente pela observação do tráfego da rede. Todos os pacotes passantes na rede são capturados e associados a um par formado pela origem e destino, tornando possível caracterizar as atividades de tráfego para um *host* em particular.

Segundo [VIE 04], entre as principais fragilidades desta ferramenta está a falta de um meio permanente de armazenamento das informações observadas, como, por exemplo, um Sistema Gerenciador de Banco de Dados (SGBD). Uma vez que a aplicação é encerrada, todos os dados são perdidos. Um outro ponto a se destacar é a não possibilidade de distribuição, pois as entidades agente e gerente formam um bloco monolítico, tornando bastante complexo o gerenciamento de múltiplos agentes distribuídos.

### 2.8.4 *NeTraMet & NeMac*

Basicamente, o *Network Traffic Meter & Network Manager Collector* (NeTraMet & NeMaC), trata-se de uma implementação do padrão proposto pelo *RTFM* do *IETF*, no contexto de medições de tráfego baseado em fluxos. Segundo [BRO 01], o *NeTraMet* foi a primeira implementação da arquitetura proposta pelo *RTFM*.

O *NeTraMet* é a entidade de *software* que implementa o medidor de tráfego, armazenando os dados dos fluxos mensurados em memória, e provendo um agente *SNMP* para tornar disponíveis as medidas para os "Leitores de Medidas" (coletores) [BRO 97a]. A distribuição *NeTraMet* inclui o pacote *NeMac* que combina uma entidade "Gerente" e uma entidade "Leitora de Medidas" (coletor). Estas duas entidades adicionais tornam possível o gerenciamento de um número arbitrário de medidores, onde cada medidor utiliza suas regras previamente estabelecidas, coletando dados de fluxos nos intervalos especificados.

Além disso, a distribuição *NeTraMet* também inclui algumas Aplicações de Análise rudimentares, permitindo aos usuários produzir gráficos simples a partir dos arquivos de

dados de fluxos disponibilizados pelo *NeMac* e para monitorar, em tempo real, os fluxos de um medidor remoto.

O *NeTraMet* é utilizado para coleta de informações em provedores de serviços de *Internet*. É bastante útil como um utilitário para entendimento do comportamento do tráfego em redes de grande amplitude. Pelo fato dos medidores de tráfego tornarem possível a especificação de regras para captura do tráfego consegue-se, ao serem estabelecidas regras precisas, uma redução significativa no volume de dados repassados aos coletores. Esta particularidade fez com que o *NeTraMet* fosse aplicado com eficiência em redes com muitos locais remotos.

## 2.9 ESTADO DA ARTE

A partir das seções previamente apresentadas, que tinham por objetivo a contextualização global acerca do tema, a presente seção visa explicitar o estado da arte no escopo deste trabalho. Inicialmente, serão traçadas as principais aproximações e conceitos que permeiam a aplicação de técnicas de amostragem a medição de tráfego convergindo, especificamente, para as proposições disponíveis no contexto de medição de tráfego baseada em fluxos, em especial na identificação de grandes fluxos (também referenciados por fluxos "Elefante") .

Embora seja um tema impulsionado mais fortemente nos últimos cinco anos e ainda com muitas lacunas a serem exploradas, os estudos e investigações sob a aplicação de mecanismos de amostragem como processo auxiliar na medição de tráfego em redes de dados, começa a ser delineado no cenário de pesquisa no trabalho apresentado em [CLA 93], no início da década de 90.

*Claffy et. al.* [CLA 93] iniciaram a investigação sobre os efeitos da implementação de mecanismos de amostragem na análise da rede em ambientes operacionais. Além do detalhamento com relação ao experimento realizado no *backbone* "*NSFNET*", os autores exploram os efeitos de diferentes parâmetros de amostragem, introduzindo conceitos que

embasam grande parte das proposições desenvolvidas na atualidade. Basicamente, são apresentadas três classes principais de métodos de amostragem: amostragem sistemática, amostragem aleatória estratificada e amostragem aleatória simples. A Figura 6 apresenta uma abstração de funcionamento para cada uma dessas classes.

Cada classe apresentada é implementada segundo um método particular, utilizando um dos dois mecanismos: baseado em evento (*event-based*) ou baseado no tempo (*time-based*). Ambos os mecanismos estão relacionados a um critério para sinalizar a seleção e inclusão de um determinado pacote na amostra, baseando-se na contagem de pacotes ou em intervalos de tempo pré-estabelecidos.

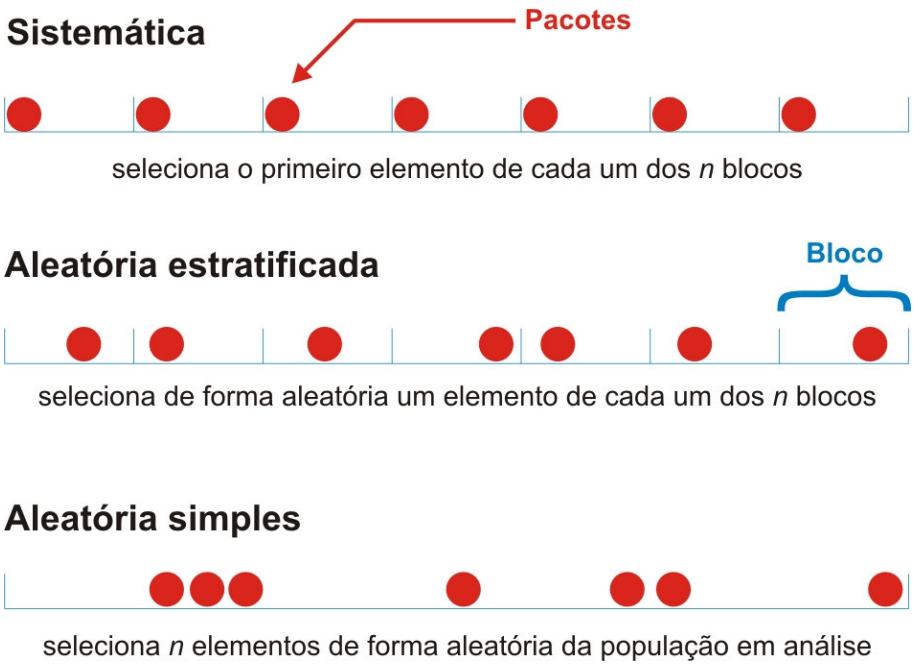



**Figura 6** Esquematização dos três algoritmos de amostragem [CLA 93].

Resumidamente, a primeira classe apresentada na Figura 6 (amostragem sistemática), realiza a seleção determinística do enésimo (*n-th*) pacote do conjunto de dados (técnica utilizada no *CISCO NetFlow* [CIS 06b]). A amostragem aleatória estratificada é similar à amostragem sistemática, exceto pelo fato que não é selecionado o primeiro pacote do bloco, uma vez que o pacote é selecionado aleatoriamente em cada bloco. Em ambas as técnicas, o tamanho do bloco não precisa ser necessariamente constante. Por último, a amostragem aleatória simples seleciona uniformemente *n* pacotes da população total, de forma aleatória.

Utilizando a combinação das diferentes classes de amostragem com os dois métodos de divisão do espaço amostral (*event-based* ou *timer-based*), *Claffy et. al*. desenvolveram um *framework* para avaliação empírica das técnicas de amostragem na caracterização do tráfego da rede, focando em duas métricas: distribuição do tamanho dos pacotes e distribuição dos intervalos de tempo entre recepção de pacotes. Resumidamente, um dos principais relatos apresentados em [CLA 93], refere-se ao fato dos resultados apontarem para um melhor desempenho nos métodos que aplicaram a seleção baseada na contagem de pacotes (*packet-triggered*, também referenciada anteriormente como *event-based*).

Ainda na década passada o departamento de monitoramento, matemática e segurança da *HP* (*Hewlett Packard*) apresentava a análise de um sistema de amostragem de pacotes, mostrando como calcular intervalos de confiança para estimar a contagem de pacotes [DUF 00]. Os autores assumem que a probabilidade de amostrar um determinado pacote contendo um tipo particular é constante durante o período de medição.

Continuamente, novas propostas relacionadas à aplicação de mecanismos de amostragem têm sido apresentadas em trabalhos científicos, denotando a possibilidade de mensuração de diferentes métricas e aplicação nos mais diversos contextos. O trabalho proposto em [DUF 00], apresenta um método direto para medição de tráfego intitulado Amostragem de Trajetória. Basicamente, a proposta baseia-se em amostrar pacotes que atravessam cada *link* (ou sub-conjunto destes *links*) de um determinado domínio autoritativo de medição. O conjunto de pacotes amostrados sob um determinado período de tempo é então usado como uma representação do tráfego completo.

A idéia chave da proposta é basear o processo de seleção da amostra em uma função *hash* determinística aplicada sobre o conteúdo do pacote. Desta forma, os autores afirmam que, se uma mesma função *hash* é utilizada para amostrar pacotes em um determinado domínio de rede, então é assegurado que um determinado pacote será selecionado em todos os *links* que atravessar, ou não será selecionado em nenhum *link*, em caso contrário. Nesse sentido, a escolha de uma função *hash* apropriada é, obviamente, crucial para garantir que o conjunto amostrado não seja tendencioso em nenhum sentido. Para tal, é destacado que o processo de amostragem, embora realizado com o uso de uma função determinística baseada no conteúdo dos pacotes, deve remeter-se a um processo de amostragem aleatória.

O trabalho apresentado em [JED 92] destaca uma nova proposta para *logging* de pacotes baseada em esquemas de *traceback* para identificação e contagem de ataques *Distributed denial-of-service* (DDoS). A idéia base da aproximação está centrada em amostrar e armazenar em *log*, um pequeno percentual de pacotes (em torno de 3.3%). Um dos principais desafios intrínsecos à utilização desta baixa taxa de amostragem, relaciona-se ao uso de técnicas mais sofisticadas a serem empregadas no processo de *traceback.* A solução apresentada constrói uma árvore de ataque usando a correlação entre os pacotes ofensivos amostrados em roteadores vizinhos. Para tal, os proponentes definem um novo esquema de amostragem que consegue garantir mais eficiência e significância em tal correlação.

A técnica de amostragem aleatória adaptativa para a identificação de alternância na carga de tráfego utilizando medições de tráfego amostradas, é apresentada em [LI 04]. Esta aproximação utiliza um erro amostral delineado com um nível de tolerância previamente estabelecido. Primeiramente, o delineamento do erro possibilita reduzir os ruídos na detecção do ponto de troca com medições de tráfego amostradas. Um segundo ponto a ser destacado na proposta relaciona-se ao fato que um nível de tolerância pré-estabelecido para o erro permite controlar o desempenho dos algoritmos de detecção de alternância de carga de tráfego assim como o número de pacotes a serem amostrados.

Basicamente, o tempo é divido (sem sobreposição) em pontos de observação (também conhecidos como blocos de tempo), sendo os pacotes amostrados em cada período de observação. No final de cada bloco, além da contabilização do volume de tráfego para o bloco, o *Squared Coeficient of Variation* (SCV) da distribuição de tamanho dos pacotes e do número de pacotes do bloco é calculado usando a amostragem do tráfego. Estes parâmetros de tráfego são utilizados para predizer o SCV da distribuição do tamanho dos pacotes e do número de pacotes para o próximo bloco, utilizando um modelo *Auto-regressive* (AR). Desta forma, a probabilidade de amostragem para o próximo bloco é determinada baseada nestes valores preditos em associação a um dado nível de tolerância de erro. A Figura 7 apresenta o processo de amostragem adaptativa proposta em [LI 04].

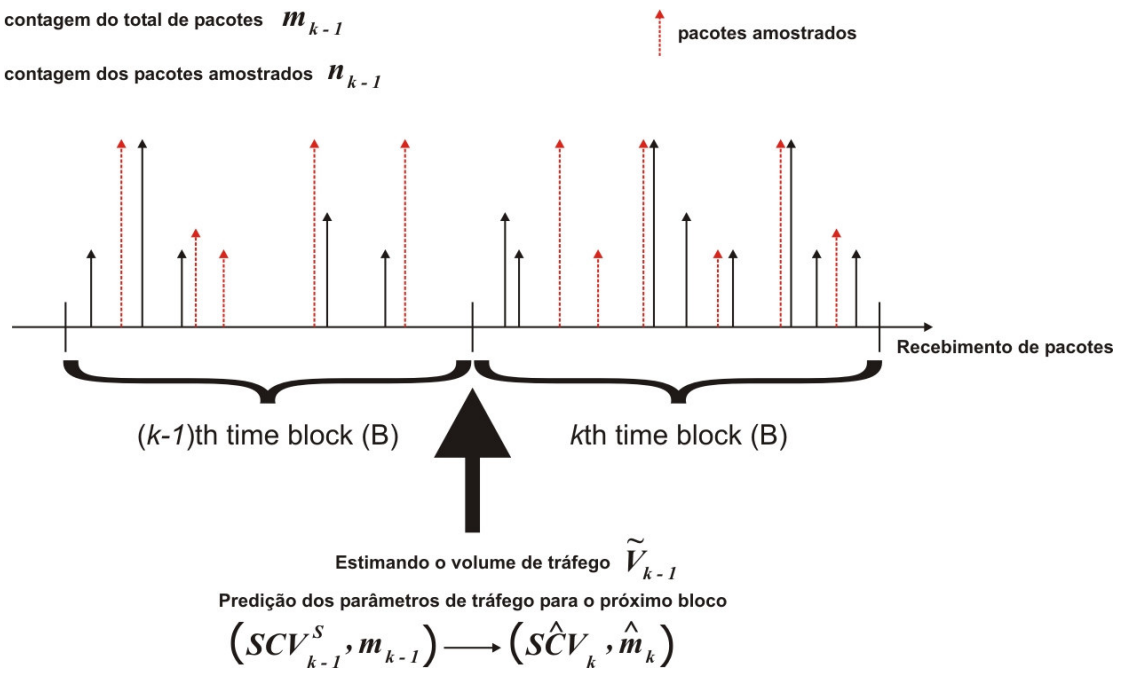



**Figura 7** Amostragem aleatória adaptativa [LI 04].

Uma avaliação relacionada à técnica de amostragem de *Poisson*, voltada especificamente a dois métodos, baseados em tempo e evento (*timer-trigger* e *event-trigger*), é apresentada em [ZHA 04]. *Zhang* e *Lei* avaliaram separadamente as duas abordagens, considerando diferentes aspectos, tais como: distribuição do comprimento dos pacotes, média do tamanho dos pacotes, além de um comparativo entre o volume de tráfego amostrado e o volume de tráfego total.

Segundo os autores [ZHA 04], a amostragem de *Poisson* aplicada com o método baseado em evento (*event-trigger*) apresenta melhores resultados de desempenho em relação à utilização com método baseado em tempo (*time-trigger*). Porém, deve ser levado em consideração que as estratégias de amostragem empregadas são estáticas, ou seja, os intervalos de coleta são configurados estaticamente. Em situações de flutuação muito acentuada das condições de tráfego, tal aproximação apresenta erros significativos, sendo necessária a aplicação de mecanismos adaptativos que garantam maior fidelidade, entre os quais destacam-se as estratégias dinâmicas sob a amostragem de *Poisson* [ZHA 04].

*Duffield et. al.* apresentam em [DUF 05] métodos que utilizam estatísticas de fluxos formadas a partir de pacotes amostrados de um determinado tráfego para inferência da freqüência do número de pacotes por fluxo no tráfego não amostrado. Desta forma, o objetivo

principal é prover mecanismos que possibilitem inferir nas propriedades dos fluxos contidos no tráfego original e que foram perdidas na população amostral. Especificamente, são apresentados os principais desafios[2] para inferência das estatísticas de fluxos a partir de pacotes amostrados e três aproximações desenvolvidas que vão ao encontro da solução de tais desafios. A primeira aproximação formaliza o argumento de escala e mostra como diluir a distribuição para predição mais acurada da distribuição de fluxos pequenos. A segunda aproximação utiliza a máxima estimativa de probabilidade para estimar a distribuição total do número de pacotes por fluxo. A terceira aproximação utiliza os detalhamentos em nível de protocolo, comumente reportados nas estatísticas de fluxos (especificamente, *flags TCP*, quando disponível) como suplemento ao comprimento de fluxos amostrados.

Um estudo analítico aprofundado sobre as três técnicas de amostragem para tráfego *Internet* auto-similar, intituladas amostragem estática sistemática, amostragem aleatória estratifica e amostragem aleatória simples (conforme apresentado na Figura 6 ), é apresentado em [HE 05]. Basicamente, é mostrado que, enquanto estas três técnicas conseguem capturar de forma acurada o parâmetro *Hurst* (estatística de segunda ordem) do tráfego *Internet*, falham na captura fiel da média (estatística de primeira ordem).

Os autores relatam que a amostragem sistemática estática modela uma menor variação nos resultados amostrados para diferentes exemplos de amostras, ou seja, conseguem obter resultados com maior fidelidade. Neste contexto, apresentam o desenvolvimento de uma nova variação da amostragem sistemática estática, intitulada *Biased Systematic Sampling* (BBS), a qual demonstra maior precisão na estimação da média, mantendo um baixo *overhead* no processo de amostragem. Finalmente, mostram que a BBS tem um ganho de desempenho de 40% e 20% (em termos de eficiência), quando comparada com as amostragens sistemática estática e aleatória simples.

Em [EST 04], é apresentada uma proposta para adaptação do mecanismo de amostragem utilizado pelo *NetFlow*. Basicamente, a proposta endereça para a adaptação dinâmica da taxa de amostragem como forma de aumentar a robustez do processo de medição, maximizando a precisão das estatísticas de fluxo. Os autores destacam que a forma como o

---

[2] No trabalho descrito em [DUF 05], são apresentados três principais desafios relacionados à inferência das informações de fluxos a partir de pacotes amostrados: estimação do número de fluxos não amostrados, as dificuldades inerentes aos fluxos pequenos e a utilização de escalas simples e suas limitações associadas.

*NetFlow* foi concebido, na qual a escolha da taxa de amostragem é realizada pelo administrador da rede e permanece estática durante todo o processo de medição, envolve alguns aspectos bastante particulares.

Quanto menor a taxa de amostragem, menor o número de pacotes a ser amostrado [EST 04]. Obviamente, isso faz com que carga do processador e o consumo de memória do dispositivo, além do uso de recursos da rede para exportação das estatísticas de fluxo, seja reduzido. Por outro lado, uma baixa taxa de amostragem ocasiona um aumento significativo do erro relativo à estimativa do tráfego real. Ainda nesse contexto, a taxa de amostragem constitui um compromisso entre duas situações opostas: quando a carga de tráfego é baixa, torna-se necessária uma taxa de amostragem maior para obter-se maior precisão, enquanto que, na ocasião de um alto volume de tráfego ou em situações de ataques massivos, torna-se necessária uma taxa de amostragem reduzida como forma de proteger a infra-estrutura de medição. Desta forma, [EST 04] os autores destacam que a decisão por parte do administrador da rede por uma taxa de amostragem ideal é extremamente complexa sendo necessário a adaptação de acordo com as condições de tráfego.

O método apresentado por *Xu et. al.* [XU 05], foca na flutuação do tráfego de redes de alta-capacidade, definindo parâmetros apropriados para descrição completa da distribuição de tráfego e então, dinamicamente, ajustar a probabilidade amostral de cada pacote, dependendo da magnitude de flutuação do tráfego. Desta forma, é observado que a granularidade é bastante fina, pois a flutuação é diretamente mapeada na probabilidade amostral de cada pacote. Existem dois pontos a serem enfatizados nesse método: a facilidade de ser instalado na rede para medição passiva; e a confiabilidade das amostras, pois suas fontes são o tráfego atual da rede em análise.

Um dos trabalhos mais atuais [HOH 06] apresenta um estudo teórico e prático sobre quais informações relacionadas ao tráfego original que podem ser inferidas quando amostras são realizadas em nível de pacotes. *Hohn et. al*. enfatizam que enquanto características básicas em nível de pacote, como estatística de primeira ordem, podem ser corretas e diretamente recuperadas, outras informações requerem maior atenção. Neste sentido, focam o trabalho, principalmente, na densidade espectral, uma estatística de segunda ordem, e na distribuição do número de pacotes por fluxo, apresentando como ambas podem ser corretamente recuperadas, na teoria. Além disso, é apresentado, detalhadamente, porque na

prática não é possível obter tais informações utilizando as amostragens tradicionais baseadas em pacotes, mesmo elevando a taxa de amostragem.

Em [IZK 06], *Izkue et. al*. propõem uma nova técnica intitulada amostragem mista. O procedimento de amostragem proposto constitui-se da combinação dos métodos baseado em evento e em tempo. Na Figura 8 são apresentados alguns pacotes em um *timeline* e alguns desses são selecionados para serem amostrados. Na amostragem baseada em evento, um a cada três pacotes é amostrado. Na baseada em tempo, o primeiro pacote do intervalo *T* é amostrado. No método proposto em [IZK 06], é utilizada uma abordagem híbrida.

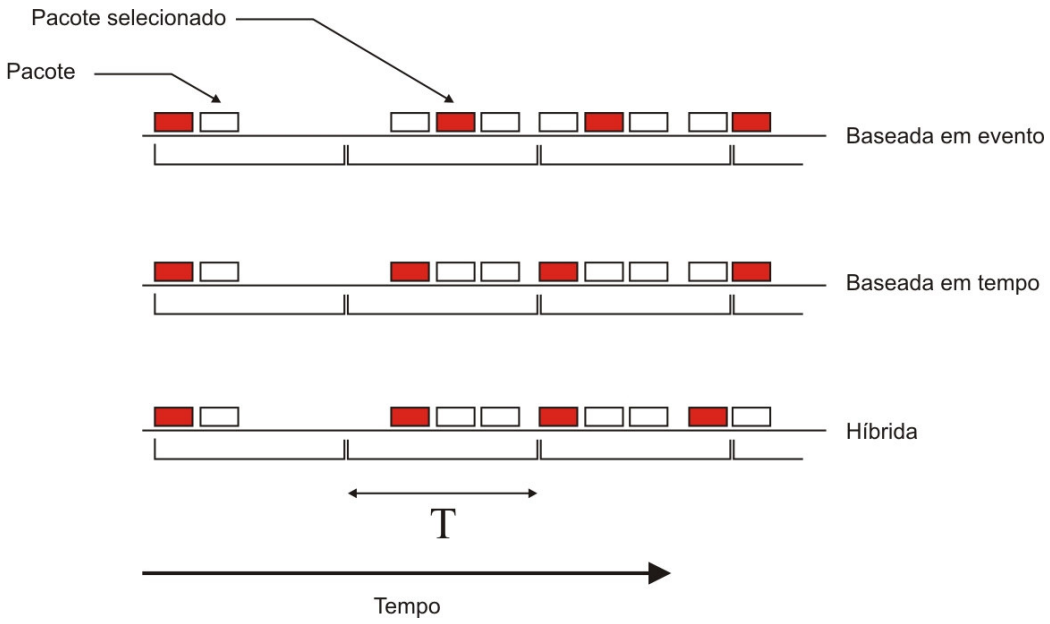



**Figura 8** Métodos de amostragem segundo [IZK 06].

A técnica estatística de amostragem estratificada foi utilizada por *Kamienski et. al.* [KAM 05] para descrição do comportamento do tráfego em nível de fluxo. O método utilizado é conhecido como amostragem estratificada ótima. Partindo de *traces* obtidos do Ponto de Presença de Pernambuco (POP-PE) da Rede Nacional de Pesquisa (RNP), os autores mostram que os resultados obtidos são representativos da população, utilizando inclusive tamanhos amostrais de 0,1% ou 0,01% do tamanho da população.

Os trabalhos previamente discutidos representam as principais iniciativas no âmbito científico acerca do tema, destacando diferentes formas de aplicação dos mecanismos de amostragem assim como estudos direcionados aos seus principais desafios e limitações,

minúcias ainda bastante em voga dada à abrangência do assunto. No contexto específico do presente trabalho, cabe destacar nesse ponto as principais propostas relacionadas à identificação de fluxos "Elefante" utilizando mecanismo de amostragem.

Basicamente, pode-se elencar três principais aproximações que norteiam a identificação de fluxos "Elefante". Ambas, focam suas expectativas baseadas na premissa do "fenômeno do Rato e do Elefante": um pequeno percentual dos fluxos, tipicamente, contabiliza o grande percentual do tráfego total [CHO 04]. Ou seja, para grande maioria das aplicações de medição e monitoramento, a estimação acurada das estatísticas de fluxos "Elefante" é, conseqüentemente, suficiente.

Inicialmente, *Estan et. al.* [EST 03] apresentam dois algoritmos para identificação de grandes fluxos. O primeiro algoritmo é intitulado "*sample and hold*" (amostrar e manter); o segundo "filtragem de múltiplos estágios". A combinação destes dois algoritmos constitui um método único para identificação de fluxos "Elefante", necessitando um baixo consumo de memória. A idéia base do algoritmo "*sample and hold*" está centrada no mecanismo de atualização das estatísticas de fluxo utilizando amostragem de pacotes. Para tal, os pacotes são amostrados com uma determinada probabilidade. Caso o pacote amostrado determine um novo fluxo (um fluxo ainda não observado), uma nova entrada na tabela de fluxos é criada. Entretanto, após uma entrada ter sido criada para um determinado fluxo todos os pacotes subseqüentes pertencentes ao fluxo são selecionados. A Figura 9 apresenta um comparativo a técnica "*sample and hold*" e o mecanismo de amostragem utilizado pelo *NetFlow*.

O complemento deste algoritmo é realizado com a "Filtragem de Múltiplos Estágios", paralelos. A idéia base está centrada no fato que um estágio é composto por uma tabela de contadores indexada por uma função *hash* computada a partir do identificador de fluxo do pacote (todos os contadores são inicializados em 0). Quando um pacote chega, a função *hash* aplicada ao seu identificador de fluxo é determinada e o tamanho do pacote é incrementado ao contador correspondente (endereçado pela função *hash*). Se todos os pacotes de um mesmo fluxo endereçam para um mesmo contador e a soma deste contador exceder um determinado limiar pré-estabelecido, tem-se a garantia de que se identificou um grande fluxo. Para minimizar falsos positivos, utiliza-se múltiplos estágios paralelos, sendo necessário que o limiar seja excedido em todos estágios para a caracterização de um grande fluxo (fluxo "Elefante"). A Figura 10 apresenta um exemplo desta aproximação.

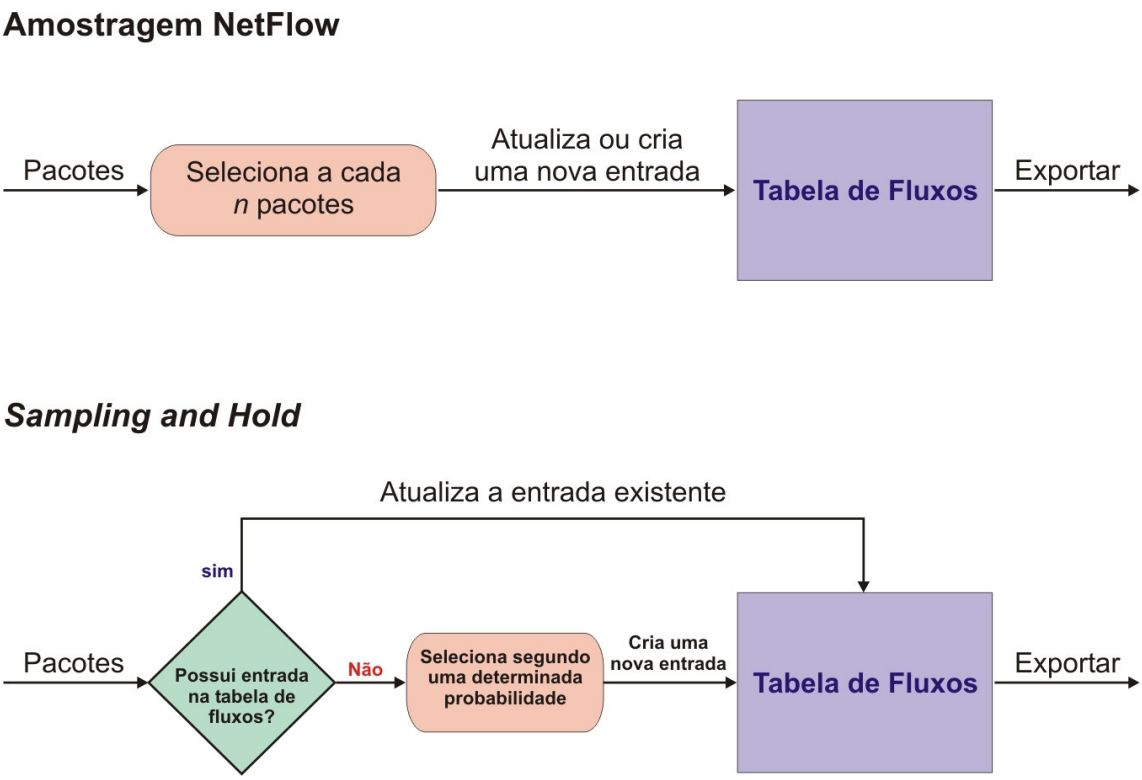



**Figura 9** *NetFlow* e o Algoritmo "*Sample and Hold*" [EST 03].

.

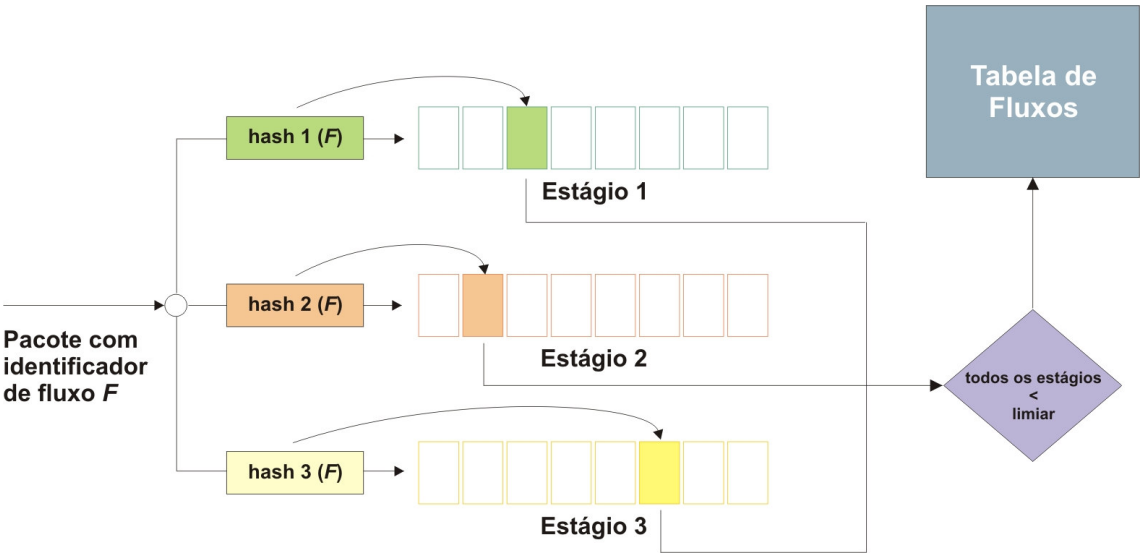



**Figura 10** Filtragem de múltiplos estágios paralelos.

Baseado no teorema de *Bayes*, o trabalho apresentado por *Mori et. al.* [MOR 04] define as técnicas e o método desenvolvido para a identificação de fluxos "Elefante" a partir da

coleta periódica de amostras de pacotes. Fundamentalmente, os autores focaram no desenvolvimento de um *framework* capaz de encontrar o limiar do número de pacotes a serem amostrados para um único fluxo, sendo essa amostra representativa a ponto de tornar possível inferência se um determinado fluxo é ou não "Elefante", considerando o tráfego original. Nessa proposta a definição quantitativa de um fluxo elefante é determinada para fluxos que contribuem em mais de 0,1% do tráfego total. O delineamento estatístico e os procedimentos para identificação de tais fluxos podem ser encontrados em [MOR 04].

Uma proposta utilizando amostragem aleatória com o uso da abordagem de estratificação, foi apresentada por *Choi et. al.* [CHO 04][CHO 06] como forma de contornar os problemas causados pela inerente dinâmica dos fluxos no tráfego de dados. Basicamente, propuseram a técnica de amostragem aleatória estratificada adaptativa para caracterização do tráfego em nível de fluxo, focando na identificação de fluxos "Elefantes". Esta aproximação foi utilizada como base para o desenvolvimento do presente trabalho e será detalhada no próximo capítulo.

### 2.9.1 PSAMP – *Packet Sampling*

O grupo de trabalho *Packet Sampling* (PSAMP) [QUI 06] foi instanciado no *IEFT* para definir um conjunto padrão de recursos para instrumentalizar elementos de rede, capacitando-os para a coleta de pacotes amostrais a partir de métodos estatísticos ou qualquer outro tipo de método pertinente. Um das premissas dessa concepção esta centrada no fato de que tais potencialidades sejam simples o bastante para que possibilitem a sua implementação de forma ubíqua em ambientes de alta-capacidade. Especificamente, o grupo de trabalho propõe-se a padronização dos seguintes mecanismos:

- seletores para amostragem de pacotes: definir o conjunto de primitivas para a operação de seleção de pacotes em elementos de rede;
- informações sobre os pacotes: especificar a extensão do pacote que deverá ser disponibilizada para relatório;

- relatórios dos pacotes amostrados: definir o formato dos relatórios que deverão ser construídos para cada pacote amostrado, para comunicação com as aplicações;
- *Report Streams*: definir o formato para relatórios de um *stream* de pacotes;
- Múltiplos *Report Streams*: definir as exigências para múltiplos seletores paralelos de pacotes em um único elemento de rede;
- configuração e gerenciamento: definir uma *MIB* para o seletor de pacotes, sendo esta alocada no elemento de rede;
- apresentação, exportação e transporte dos relatórios de pacotes: definir a interface para apresentação dos relatórios para as aplicações.

Embora este grupo de trabalho ainda não tenha padronizado suas iniciativas em nível de *RFC*, a *draft* "*Sampling and Filtering Techniques for IP Packet Selection*" [ZSE 05], possui um grande conjunto de diretivas que impulsionam os estudos e investigações sobre a utilização de mecanismos de amostragem na medição de tráfego em redes de comunicação de dados. Outras *draft´s* apresentadas pelo *PSAMP* direcionam para os demais aspectos que são escopo do grupo de trabalho, como: especificação do protocolo *PSAMP* [CLA 06], modelo de informação para exportação dos pacotes amostrados [DIE 06] e o *framework* para seleção de pacotes e relatórios.

Em [ZSE 05], é apresentada a categorização das técnicas de seleção de pacotes. Basicamente, as técnicas de seleção de pacotes geram um subconjunto de pacotes a partir de um fluxo de dados observado em um determinado ponto de observação, sendo divididas em duas categorias: amostragem e filtragem.

A amostragem no contexto do *PSAMP* tem por objetivo, a seleção de um conjunto significativo de pacotes e este conjunto é utilizado para inferir o comportamento completo sem a necessidade de analisar todos os pacotes. A seleção pode utilizar como critério a posição do pacote e/ou baseada no conteúdo do pacote e/ou baseada em decisões aleatórias.

A filtragem seleciona um conjunto com propriedades comuns, sendo apropriada apenas se é desejado analisar um grupo de pacotes de interesse. Tais propriedades podem ser diretamente derivadas do conteúdo do pacote ou dependente do tratamento dado pelo roteador

ao pacote. A filtragem é caracterizada como uma operação determinística, dependendo do conteúdo do pacote ou do tratamento do roteador, não sendo jamais dependente da posição do pacote ou de decisões aleatórias [ZSE 05]. Resumidamente, a Tabela 5 relaciona os esquemas de seleção considerados pelo *PSAMP* com as categorias de seleção (amostragem e filtragem), indicando as propriedades de cada um dos esquemas.

**Tabela 5** Resumo dos esquemas de seleção abordados pelo PSAMP [ZSE 05]

| Esquema de seleção | 1 | 2 | 3 | A | F |
|---|---|---|---|---|---|
| Sistemática baseada em contagem | X | - | - | X | |
| Sistemática baseada em tempo | X | - | - | X | |
| Aleatória *n-out-of-N* | - | - | - | X | |
| Aleatória com probabilidade uniforme | - | - | - | X | |
| Aleatória com probabilidade não uniforme | - | X | X | X | |
| Aleatória não uniforme baseada em estado de fluxo | - | X | X | X | |
| Filtragem baseada em coluna | X | X | - | | X |
| Função *hash* | X | X | - | | X |
| Filtragem baseada em rota | X | X | X | | X |

Critérios de Seleção: 1 - Seleção determinística; 2 - Variáveis dependentes do conteúdo; 3 - Variáveis independentes do conteúdo. Categorias de Seleção: A – Amostragem; F - Filtragem

A descrição detalhada de cada esquema de seleção apresentado no Tabela 5 pode ser obtida em [ZSE 05]. Embora não estejam diretamente relacionados ao presente trabalho, os esforços e iniciativas traçadas pelo grupo de trabalho *PSAMP*, denotam a fertilidade do tema e impulsionam grande parte dos trabalhos publicados nesta área. Por este motivo, consolida-se como uma das fontes elementares para entendimento das proposições associadas à aplicação de mecanismos de amostragem na seleção de pacotes, assim como as demais funcionalidades requeridas e intrínsecas a esse processo (exportação, transporte, relatórios, etc).

Além disso, a constante reciclagem deste grupo de trabalho com a publicação de novas *draft´s*, norteando cada vez mais para uma suposta padronização, realimenta o tema e consolidasse naturalmente como umas das principais referências do estado da arte.

# 3 AMOSTRAGEM ALEATÓRIA ESTRATIFICADA ADAPTATIVA NA IDENTIFICAÇÃO DE FLUXOS "ELEFANTE"

O presente capítulo tem por objetivo apresentar a técnica de amostragem que foi implementada e avaliada, no contexto de identificação de fluxos "Elefante". Para tal, este descritivo foi embasado nos trabalhos publicados em [CHO 04] e [CHO 06]. O delineamento estatístico de algumas definições que embasam a técnica usada pode ser analisado nos anexos do trabalho, os quais serão referenciados no decorrer da explanação.

Inicialmente, tratar-se-á da questão relacionada à identificação de um fluxo "elefante". Na técnica de amostragem em questão, um fluxo "elefante" é determinado em termos da contabilização de pacotes (número de pacotes), ou seja, a decisão de armazenar ou descartar um fluxo não é realizada com base em propriedades do pacote, como por exemplo, o tamanho em *bytes*.

Empiricamente, a adoção deste critério tem como base o fato de que um fluxo que contabiliza uma quantidade suficientemente grande de pacotes, também contabiliza uma quantidade proporcional de *bytes* e vice-versa. Isto efetivamente ocorre devido o tamanho máximo dos pacotes ser limitado pela *Maximum Transmission Unit* (MTU) da rota percorrida. Desta forma, grande fluxos, em termos de contabilização em *bytes*, invariavelmente, sofrem fragmentação, gerando um número proporcional de pacotes.

Partindo-se desta constatação, os fluxos podem ser classificados como "elefante" baseando-se na proporção da contabilização de pacotes no intervalo de tempo que abrange a total duração do fluxo. Ou seja, um fluxo é classificado como "elefante" se a proporção de pacotes contabilizados para este fluxo excede um limiar pré-estabelecido, por exemplo, 1 % do tráfego total (tráfego total observado durante o intervalo de duração do fluxo). A Figura 11 ilustra esta definição.

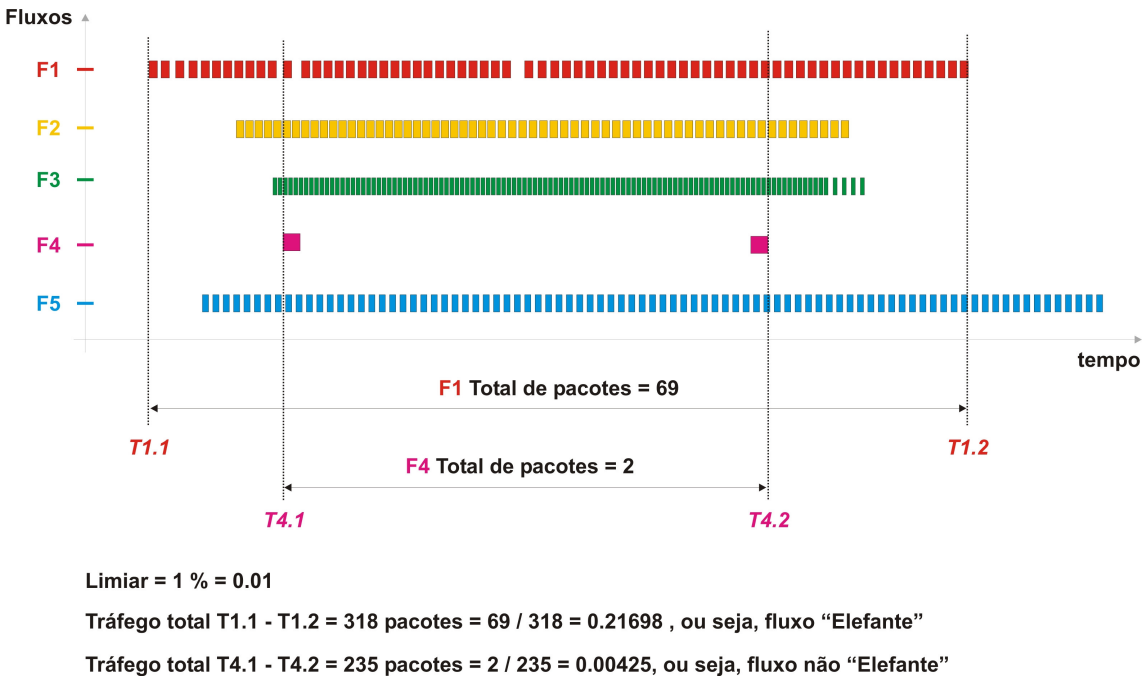



**Figura 11** Exemplo da definição de fluxo “elefante”.

Com base nesta definição para determinação de um fluxo “elefante” e com o requisito de manter precisão e confiabilidade na aplicação de algum mecanismo de amostragem, os autores identificam a necessidade de determinar um intervalo de amostragem, para o qual a taxa de amostragem (seleção de amostras) é ajustada em conformidade com as modificações nas condições do tráfego. Entretanto, existe uma dificuldade intrínseca na definição de um intervalo ótimo. Esta dificuldade surge devido à dinâmica de chegada e duração dos fluxos o que faz com que seja muito difícil inferir um intervalo de amostragem que seja válido, pois o mesmo deve considerar diferentes perfis de tráfego compreendendo a dinâmica de quaisquer fluxos “elefantes”.

Para contornar esta dificuldade, a técnica de amostragem utiliza a aproximação de estratificação no tempo. Neste contexto, o método de amostragem estratificada é aplicado dividindo-se o tempo (período de medição) em intervalos pré-determinados e não sobrepostos, chamados de estratos ou blocos de tempo. Esta aproximação está ilustrada na Figura 12 .

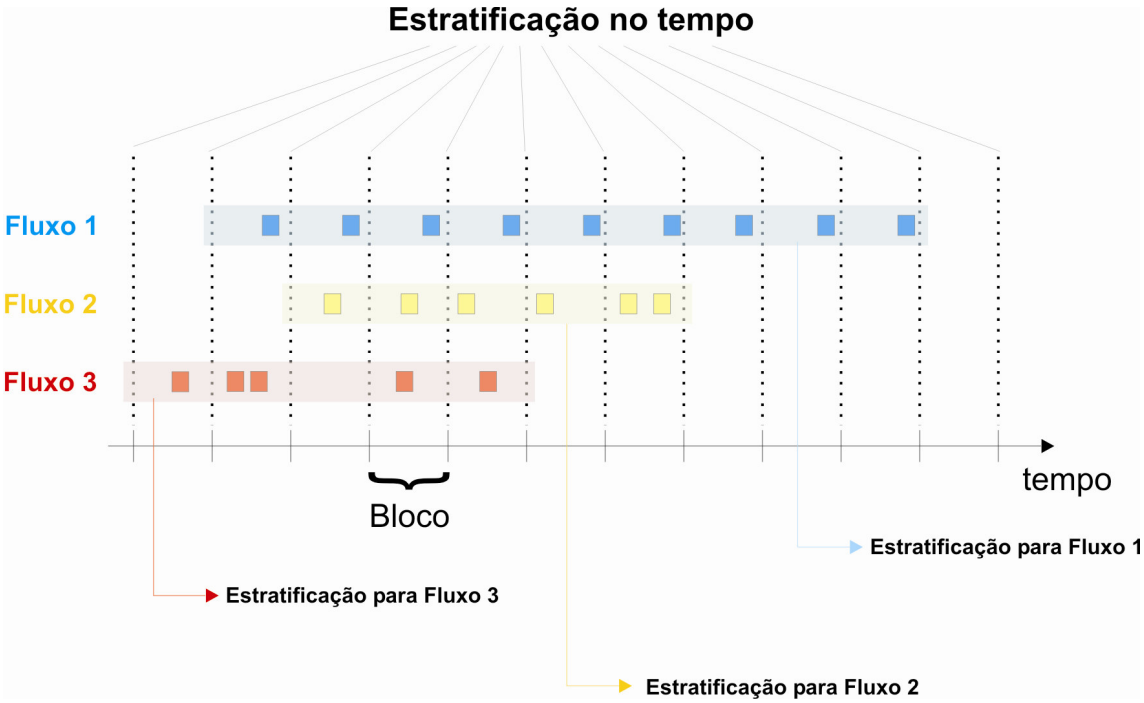



**Figura 12** Estratificação no tempo do intervalo de medição.

Assim, para cada bloco de tempo os pacotes são amostrados com a mesma probabilidade, ou seja, utilizando, simplesmente, amostragem. No final de cada bloco, as estatísticas dos fluxos são estimadas e atualizadas. A partir daí, o volume final do fluxo é sumarizado no fechamento do último bloco de tempo, em que o fluxo foi observado. Sob o ponto de vista de cada fluxo, a sua duração é dividida ou estratificada em um tempo fixo, permitindo a estimativa de seu volume, sem conhecer a dinâmica de chegada e de duração do fluxo.

Partindo de tais premissas, a definição formal de um fluxo elefante é então formalizada, como segue: “considerando-se um intervalo de tempo quantificado que contenha a duração completa de um fluxo $f$. Supondo-se, ainda, que este intervalo seja constituído de $L$ blocos consecutivos, onde um total $m_i$ de pacotes são observados no bloco $i$ ($i$ = 1 ... $L$) e $m^f$ os pacotes pertencentes ao fluxo $f$ dentre o total de pacotes $m$, então, diz-se que um fluxo é “elefante”, equação (1), se a proporção da contabilização de pacotes para o fluxo $p^f$ é maior que um limiar $p^\theta$ pré-estabelecido”.

$$\frac{m^f}{m} = \frac{\sum_{i=1}^{L} m_i^f}{\sum_{i=1}^{L} m_i} = p^f \geq p^{\theta} \qquad (1)$$

Com base nesta definição, a identificação ou não de um fluxo "elefante" pode ser realizada apenas mantendo um contador do total de pacotes observados nos blocos em que o fluxo perdurou. Quando um fluxo expira, a proporção da contabilização do total de pacotes para o fluxo, sobre o total de pacotes observados durante os blocos, indica diretamente se o fluxo é ou não elefante, conforme mencionado anteriormente.

A partir deste ponto, para cada fluxo "elefante" identificado, torna-se necessário fazer à estimativa da contabilização do total de pacotes $\hat{m}^f$ e a estimativa da contabilização do total de *bytes* $\hat{v}^f$ para o fluxo. Para tal, é preciso limitar o erro relativo da estimativa. Ou seja, dado um nível de tolerância de erro pré-estabelecido, $\{\eta, \varepsilon\}$, (onde (1 - $\eta$) e $\varepsilon$ referem-se à confiabilidade e precisão, respectivamente, e $0 \leq \eta \leq 1$), a estimação de erro para a contabilização de pacotes e *bytes* deve ser limitada, de acordo com a seguinte expressão:

$$\Pr\left\{\left|\frac{\hat{m}^f - m^f}{m^f}\right| > \varepsilon\right\} \leq \eta \text{ e } \Pr\left\{\left|\frac{\hat{v}^f - v^f}{v^f}\right| > \varepsilon\right\} \leq \eta \qquad (2)$$

Onde $p^f \geq p^{\theta}$ para o fluxo $f$. Com base neste delineamento inicial, a seção seguinte foca na apresentação de como a técnica de amostragem determina o número mínimo de amostras requeridas em um estrato (bloco de tempo) dentro de um limite de erro pré-estabelecido, objetivando a estimativa do número de pacotes e do volume de *bytes*. Além disso, será apresentada a aplicação da abordagem de predição com o uso de um modelo Auto-Regressivo para o estabelecimento da probabilidade de amostragem.

### 3.1.1 Determinação do número requerido de amostras

Conforme já mencionado, o método de amostragem é aleatório. Considerando-se, ainda, que:

- o tamanho da amostra, dada por $n$, é suficientemente grande ($n > 30$);
- o tamanho da população, dada por $m$ (população), é muito grande quando comparada com o tamanho da amostra $n$: $m >> n$, de tal forma que a fração de amostragem seja pequena.

Sob estas condições, o Teorema do Limite Central (TLC) é aplicável, ou seja, a distribuição amostral da média tem distribuição normal com média $\mu$ e desvio padrão $\frac{\sigma}{\sqrt{n}}$, independentemente da distribuição da população, sendo $\mu$ e $\sigma$ a média da população e o desvio padrão da população, respectivamente. É conveniente destacar que as amostras são independentes e identicamente distribuídas (i.i.d), conforme a condição imposta pelo TLC e obtidas de forma simples, utilizando-se basicamente uma amostragem aleatória a partir de uma população comum.

Assim, utilizando amostragem aleatória, a contagem de pacotes para um fluxo é estimada como segue:

- considera-se um intervalo unitário de tempo que contenha a duração completa de um fluxo $f$, no qual $m$ pacotes são observados;
- a partir dos $m$ pacotes observados, $n$ pacotes são aleatoriamente amostrados ($n < m$), resultando em $n^f$ pacotes pertencentes ao fluxo $f$.

Assim, a contabilização de pacotes para o fluxo $f$, $m^f$, é estimada por $\hat{m}^f$ utilizando a proporção de amostragem $\hat{p}^f$, conforme a equação apresentada a seguir:

$$\hat{m}^f = m.\frac{n^f}{n} = m.\hat{p}^f \tag{3}$$

A proporção pode ser considerada como um caso especial da média aritmética, onde uma variável $I$ possui apenas os valores 0 e 1. Por exemplo, supondo que se deseje encontrar a proporção para um fluxo particular $f$, considerando que existam $m$ pacotes e que $I_i = 1$ se o *i-ésimo* pacote pertence ao fluxo $f$ e $I_i = 0$, caso contrário. Então o número de pacotes pertencentes ao fluxo $f$ é $m^f = \sum_{i=1}^{m} I_i$. A proporção de pacotes para o fluxo é calculada para contabilização do total de pacotes durante o intervalo:

$$p^f = \frac{m^f}{m} = \frac{\sum_{i=1}^{m} I_i}{m} \tag{4}$$

Considerando que $\hat{I}_1, \hat{I}_2, ..., \hat{I}_n$ são $n$ amostras aleatórias e $n^f$ o número de pacotes amostrados pertencentes ao fluxo $f$, a proporção de amostragem do fluxo $f$ é então determinada como segue:

$$\hat{p}^f = \frac{n^f}{n} = \frac{\sum_{j=1}^{n} I_j}{n} \tag{5}$$

No interior de um bloco de tempo (estrato), os pacotes são amostrados com probabilidade fixa (igualmente prováveis). Assim sendo, de acordo com o Teorema do Limite

Central (TLC) para amostras aleatórias[3] quando $n \to \infty$, a média da amostra $\hat{p}^f$ se aproxima da média da população $p^f$ e a variância de $\sigma^2_{\hat{p}^f} = \frac{p^f \cdot (1 - p^f)}{n}$ (ver ANEXO A), independentemente da distribuição da população. Desta forma, a proporção amostral pode ser escrita em termos da média e variância:

$$\hat{p}^f \approx p^f + \frac{\sqrt{p^f \cdot (1 - p^f)}}{\sqrt{n}} \cdot Y_P = \mu_P + \sigma_P \cdot Y_P \tag{6}$$

Onde $Y_p \sim N(0,1)$ é a função de densidade de probabilidade padronizada para a estimativa da contagem de pacotes. A equação (2) pode ser reescrita da seguinte forma:

$$\begin{aligned} & P\left\{\left|\frac{m\hat{p}^f - mp^f}{mp^f}\right| > \varepsilon\right\} = P\left\{\left|\frac{\hat{p}^f - p^f}{\sigma_{\hat{p}^f}}\right| > \frac{p^f \sqrt{n}\varepsilon}{\sqrt{p^f (1 - p^f)}}\right\} \\ & \approx 2\left(1 - \Phi\left(\frac{\sqrt{p^f}\sqrt{n}\varepsilon}{\sqrt{(1 - p^f)}}\right)\right) \leq \eta \end{aligned} \tag{7}$$

Onde $\Phi(\cdot)$ é a função de distribuição cumulativa (c.d.f. - *cumulative distribution function*) da distribuição normal padrão. Resolvendo a desigualdade da equação (7) explicitando-se $n$, é então possível derivar o número mínimo de amostras requeridas $n^{*,p}$ para estimar a contabilização de pacotes dentro de um nível pré-estabelecido de tolerância para o erro $\varepsilon$:

---

[3] TEOREMA DO LIMITE CENTRAL - Se $X_1, X_2, \ldots, X_n$ são variáveis aleatórias independentes com média $\mu$ e variância $\sigma^2$, a distribuição de $Y = \sum_{i=1}^{n} X_i$, se torna aproximadamente igual a uma distribuição normal com média $E(Y) = n.\ \mu$ e variância $V(Y) = n.\ \sigma^2$, a medida em que "$n$" cresce.

$$n \geq n^{*,p} = \left[ z_p \cdot \left( \frac{1 - p^f}{p^f} \right) \right] = \left[ z_p \cdot C_\theta \right] \quad (8)$$

Onde $z_p = \left( \frac{\Phi^{-1}(1-\eta/2)}{\varepsilon} \right)^2$ e $C_\theta = \frac{1-p^\theta}{p^\theta}$ são constantes. Desta forma, com um número aleatório mínimo de $n^{*,p}$ amostras, uma amostragem aleatória simples pode fornecer uma confiabilidade pré-especificada $\{\eta, \varepsilon\}$ para qualquer proporção de fluxo que seja maior que um limite estabelecido, $p^\theta$, considerado como fluxo "elefante".

### 3.1.2 Estimação do volume de *bytes*

O delineamento apresentado na seção 3.1.1, demonstra como obter o número mínimo de amostras requeridas para a estimação do total de pacotes que pertencem a um dado fluxo $f$, identificado como "elefante". Nesta seção será apresentado o delineamento para determinar o número mínimo de amostras necessárias para que possa ser realizada, adicionalmente, a determinação do volume de *bytes* para o fluxo $f$. Cabe salientar que, no contexto do presente trabalho, o número de amostras foi calculado a partir das especificações detalhadas nessa seção, pois se deseja contabilizar tanto o número de pacotes como o volume de *bytes*, para os fluxos "elefante".

Inicialmente, se considera que a mensuração do volume de *bytes* para um fluxo $f$ é dada pela seguinte fórmula: $v^f = m^f \mu^f$, onde $\mu^f$ é a média do tamanho dos pacotes pertencentes ao fluxo $f$. Similarmente, a mensuração estimada do volume de *bytes* para o fluxo $f$ pode ser escrita da seguinte forma:

$$\hat{v}^f = \hat{m}^f \cdot \hat{\mu}^f = m \cdot \hat{p}^f \cdot \hat{\mu}^f \tag{9}$$

Onde $\hat{\mu}^f$ é a média estimada do tamanho dos pacotes pertencentes ao fluxo $f$. Segundo [CHO 04][CHO 06], a equação (9) indica dois pontos de incerteza que envolvem a estimação do volume de *bytes*: a estimação de proporção do fluxo ( $\hat{p}^f$ ) e a estimação da média do tamanho de pacotes para o fluxo ( $\hat{\mu}^f$ ). Para eliminar tais pontos de incerteza recorre-se a dois lemas: a consistência da proporção da amostra e uma extensão do TLC para uma soma de um número aleatório de variáveis aleatórias. Estes dois lemas são apresentados no ANEXO B.

Aplicando esses dois lemas, a estimação do volume de *bytes* pode ser aproximada com a soma de duas variáveis aleatórias normais, conforme a equação (10):

$$\hat{v}^f = mp^f \mu^f + m\left[\frac{\sqrt{p^f}}{\sqrt{n}}(\mu^f \sqrt{1-p^f}\, Y_p + \sigma^f Y_b)\right] \tag{10}$$

Onde $Y_b, Y_p \sim N(0,1)$.

Assim, o número de amostras requeridas para estimação do volume de *bytes* para um determinado fluxo pode ser obtida de forma similar a estimação da quantidade de pacotes, conforme a equação (11):

$$n \geq n^{*,b,f} = \left[ z_p . \left( \frac{1-p^f + S^f}{p^f} \right) \right] \tag{11}$$

Onde $S^f = \left( \frac{\sigma^f}{\mu^f} \right)^2$ é o SCV (quadrado do coeficiente de variação) do tamanho dos pacotes pertencentes ao fluxo $f$. De acordo com os autores, embora a variabilidade do tamanho dos pacotes (SCV) nos fluxos seja geralmente grande (variando entre 0,00007 a 8), para fluxos "elefante" esta variabilidade é bastante limitada. Ou seja, fluxos grandes tendem a ter pacotes com tamanhos similares. Desta forma, pode-se, eficientemente, estipular um limite razoável de SCV para fluxos "elefantes", em torno de 0,2(< 1), por exemplo. Conseqüentemente, o número de amostras requeridas, dentre um limite de erro pré-estabelecido, para a estimação do volume de *bytes* pode ser escrita da seguinte forma:

$$n \geq n^{*,b} = \left[ z_p . B_\theta \right] \tag{12}$$

Onde $B_{\theta=} \left( \frac{1 - p^\theta + S^\theta}{p^\theta} \right)$.

3.1.3 Probabilidade de amostragem e predição do total de pacotes

Tendo o número requerido de amostras computadas, a probabilidade para um bloco produzir $n^*$ ($n^{*,p}$ ou $n^{*,b}$, no caso do presente trabalho utilizou-se $n^{*,b}$, como mencionado anteriormente) amostras é:

$$P_{sp} = \frac{n^*}{m_h} \tag{13}$$

Onde $m_h$ é o número total de pacotes pertencentes ao bloco *h*. O termo $n^*$ pode ser $n^{*,p}$ para estimar apenas a quantidade de pacotes, ou $n^{*,b}$, para o caso de estimar, adicionalmente,

o volume de *bytes*. Entretanto, não se consegue escolher por uma taxa de amostragem precisa quando o tamanho da população (contagem total de pacotes para um bloco de tempo) é desconhecido. Para solucionar este problema, [CHO 04][CHO 06] utilizaram uma predição no início de um novo bloco para computar a taxa de amostragem.

Especificamente, foi utilizado um modelo Auto-Regressivo AR(1) (ver ANEXO C) de série temporal para predizer o tráfego total em termos do número de pacotes ($m$), para o próximo bloco, utilizando 5 valores passados (neste caso, os valores passados referem-se ao total de pacotes observados nos últimos $n$ blocos). A Figura 13 ilustra este procedimento.

Para finalizar o processo, a atualização das estatísticas dos fluxos ativos utilizando a taxa de amostragem no final do bloco $h$ é dada como [CHO 06]:

$$\hat{m}_h^f = \hat{m}_{h-1}^f + \frac{m_h}{n_h}\hat{n}_h^f, \quad \hat{v}_h^f = \hat{v}_{h-1}^f + \frac{m_h}{n_h}\hat{n}_h^f\hat{\mu}_h^f \tag{14}$$

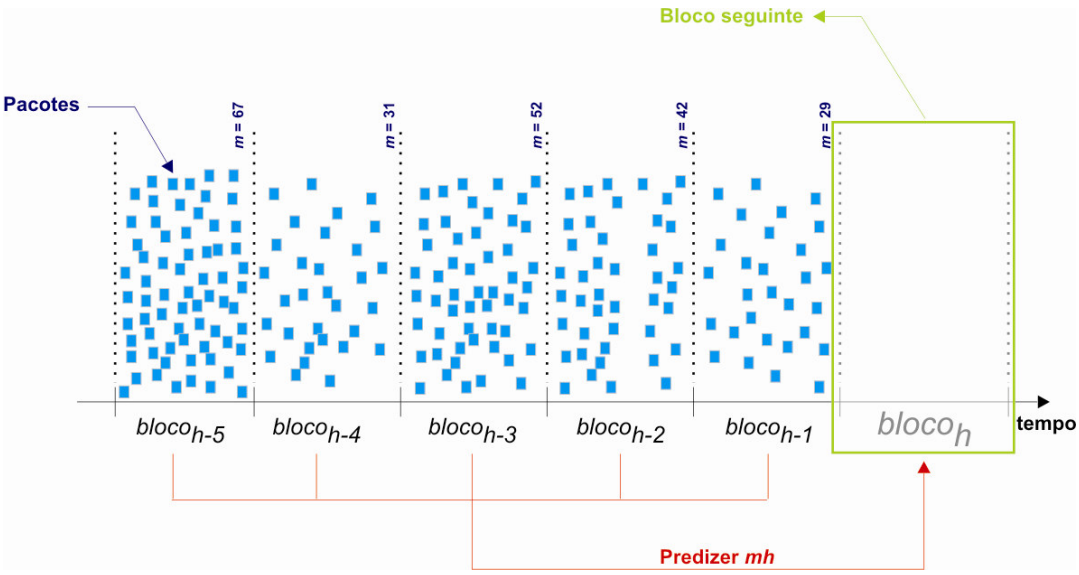



**Figura 13** Ilustração do mecanismo de predição.

# 4 DESENVOLVIMENTO DO PROTÓTIPO

No presente capítulo são apresentadas as tecnologias utilizadas para o desenvolvimento do trabalho proposto. Adicionalmente, é apresentado o desenvolvimento do protótipo propriamente dito, e os ambientes de teste utilizados para avaliação dos resultados.

## 4.1 ARRANJO EXPERIMENTAL

Esta seção tem por objetivos apresentar as tecnologias utilizadas para o desenvolvimento do presente trabalho, destacando: sistema operacional, linguagem de programação, algoritmo para geração de números aleatórios, sistema para armazenamento das estatísticas de fluxos (*PostGreSQL*) e a biblioteca para captura passiva do tráfego (*Libpcap*).

### 4.1.1 Sistema Operacional

Como etapa preponderante no desenvolvimento do presente trabalho, destacou-se a definição do Sistema Operacional (SO) adequado e das tecnologias associadas. Ambos os recursos devem estar em conformidade com as definições e padrões utilizados no GPARC&TI e, adicionalmente, contemplar os requisitos funcionais requeridos pelo protótipo a ser desenvolvido.

Primeiramente, optou-se pela adoção de algum dos SO´s da família *GNU/Linux*, pois a grande maioria dos sistemas, ferramentas e bibliotecas desenvolvidas pelo GPARC&TI utilizam como base estes SO´s, fato que caracteriza, paralelamente, a familiaridade e *expertise* dos pesquisadores com este recurso. Outro ponto de interesse no emprego desta categoria de SO´s relaciona-se ao fato destes proporcionarem muita flexibilidade, pois são distribuídos com código-fonte aberto, possibilitando a alteração e adequação de seu funcionamento de

acordo com necessidades de interesse. Além disso, a família *GNU/Linux*, a cada dia, soma mais espaço tanto no mundo acadêmico como no empresarial, apresentando soluções com a eficiência requerida e com baixo custo, viabilizando economicamente o desenvolvimento de novas soluções.

### 4.1.2 Linguagem de programação

A partir da seleção do SO, foi efetuada à escolha da linguagem de programação. Conseqüentemente, tornou-se necessário optar por uma linguagem que estivesse em alinhamento com o SO e possibilitasse a manipulação de recursos em baixo nível, pois o cerne do aplicativo é composto por um conjunto de *software*s básicos. Além disso, a linguagem deve garantir facilidades na integração com outros tipos de *interface*, em especial com *Application Programming Interface* (API) para interação com os dispositivos de rede e com bibliotecas externas.

Desta forma, a escolha pela linguagem para o desenvolvimento do trabalho proposto deu-se naturalmente, optando-se pela linguagem de programação *C++*. Segundo [SCH 98], o *C++* surge no cenário de desenvolvimento como uma necessidade por um dos principais fatores relacionados à programação: o aumento de complexidade. Neste sentido, aparece como uma linguagem potencialmente adaptada para o desenvolvimento de sistemas maiores e mais complexos, através da incorporação dos conceitos da Programação Orientada a Objetos (POO), fato que possibilita ainda o alinhamento com as principais tendências no contexto da Engenharia de *Software*.

Outro ponto de interesse na escolha desta linguagem de programação está relacionado à questão de portabilidade. A partir da padronização realizada no comitê ANSI/ISO C++ e completada em 14 de novembro de 1997 [SCH 98], grande parte dos principais compiladores passaram a estar de acordo com as recomendações. Desta forma, a portabilidade dos sistemas torna-se simplificada, possibilitando uma ágil adaptação para outras arquiteturas e sistemas operacionais.

Outras *features* também fomentam a tendência pelo uso da linguagem C++. Um recurso de grande utilidade é a *Standard Template Library* (STL), desenvolvida por *Alexander Stepanov* (1997). Basicamente, a STL é um conjunto de rotinas genéricas que podem ser utilizadas para a manipulação de dados. Fornece funcionalidades de propósitos gerais, classes *templates* e funções que implementam os mais comuns e populares algoritmos e estruturas de dados, incluindo, por exemplo, suporte a vetores, listas, filas e pilhas. Em http://www.cppreference.com/, é apresentada uma referência completa a STL e demais funcionalidades da linguagem, através de exemplos práticos complementados dos conceitos relacionados.

### 4.1.3 Algoritmo para geração de números aleatórios

Outra necessidade imposta para a realização do trabalho, esteve centrada na utilização de um algoritmo para geração de números aleatórios, também referenciados por *Random Number Generation* (RNG). A técnica de amostragem proposta em [CHO 04] e [CHO 06], avaliada no presente trabalho, não apresenta recomendação sobre qual algoritmo deve ser aplicado para aleatorizar à seleção de amostras nos estratos. Por este motivo, o primeiro passo antes de decidir por um algoritmo específico, fez-se contato via *e-mail* com os autores da proposta, os quais retornaram informando que poderia ser utilizada qualquer função para geração de números aleatórios disponibilizada pela linguagem de programação utilizada (ver APÊNDICE A).

A partir da resposta dos autores, constatou-se que poderia ser utilizada diretamente a função "*Rand()*", disponibilizada pelo C++. Porém, esta função não possibilita a geração nativa de números aleatórios em um determinado intervalo (*1 ... N*). Neste sentido, optou-se por utilizar a implementação do algoritmo desenvolvido por *Makoto Matsumoto* e *Takuji Nishimura* [MAT 98], intitulado *Mersenne Twister*. Esta implementação foi realizada em C++ mantendo compatibilidade com os demais módulos, além de ser distribuída com licença *freeware*. Entretanto, a geração de números aleatórios mantém-se como um dos pontos de interesse na avaliação da técnica de amostragem utilizada.

### 4.1.4 *PostGreSQL*

Devido à necessidade de armazenamento dos dados de fluxo coletados pelo sistema proposto, tornou-se imprescindível à utilização de um meio robusto, rápido e com alto nível de confiabilidade. Estas características são requeridas pela natureza e pelas particularidades intrínsecas relacionadas à posterior análise dos dados coletados. Cada vez mais, as ferramentas de monitoramento e gerenciamento de redes tendem a traçar o comportamento com base em dados históricos (*baseline*), guardar registros de informações para casos de quebras de SLA, exibição de relatórios passados, entre outros.Para suprir tal necessidade optou-se por utilizar o SGBD *PostGreSQL.*

Os sistemas gerenciadores de banco de dados surgiram no início da década de 70 com o objetivo de facilitar a programação de aplicações de banco de dados [HEU 00]. Neste contexto, o *PostGreSQL* destaca-se como um SGBD objeto-relacional com código aberto. Ele é considerado objeto-relacional por implementar, além das características de um SGBD relacional, algumas características de orientação a objetos, como herança e tipos personalizados.

Adicionalmente, o trabalho desenvolvido por [SAN 07a] propõe um sistema para exportação das informações de fluxo, com base nas recomendações do IPFIX, no qual foi realizada uma adaptação da biblioteca *libIPFIX*, apresentada por [MAR 06]. Especificamente, a *libIPFIX* foi portada (nativamente esta biblioteca utiliza *mySQL*) permitindo que a entidade "Coletor IPFIX" (ver Figura 4 ) armazene as estatísticas de fluxos em um SGBD *PostGreSQL.* Este fato consolidou-se como um outro fator preponderante para a escolha deste gerenciador, aliado aos demais benefícios elucidados anteriormente. Outro ponto a ser levado em consideração relaciona-se ao *PostGreSQL*, a partir de sua versão 8.0, possuir distribuição nativa para o sistema operacional *Windows*, fato que torna ainda mais viável a portabilidade do sistema implementado.

### 4.1.5 Biblioteca *Libpcap*

Devido à natureza do trabalho proposto, tornou-se necessário utilizar mecanismos que possibilitassem a coleta do tráfego passante (captura passiva) em um ponto de interesse do ambiente de rede em análise. Especificamente, é preciso capturar cada datagrama que trafega pelo *link* de interesse, para que desta forma os fluxos sejam detectados e suas estatísticas contabilizadas (isso em ambos os casos, tanto na abordagem tradicional como na abordagem utilizando amostragem de pacotes).

Para tal utilizou-se a biblioteca *libpcap* (versão *Unix-like* da *Pcap*). Esta biblioteca é amplamente usada por ferramentas consagradas que necessitam do recurso de captura passiva, tais como: *tcpdump*, *wireshark* (antigo *ethereal*), *snort*, entre outros. Basicamente, a *Libpcap* é uma abstração em alto nível, contendo um conjunto de funções para manipular a *Pcap*. Por outro lado, a *Pcap* é a responsável pela captura propriamente dita dos pacotes, permitindo a utilização de filtros *Berkeley Packet Filter* (BPF) [MCC 93]. A *Pcap* foi desenvolvida por *Van Jacobson*, *Craig Leres* e *Steven McCanne*, todos do *Lawrence Berkeley National Laboratory*, Universidade da Califórnia, *Berkeley*, CA.

A *Libpacap* permite definir a forma como os pacotes serão capturados, operando de duas formas distintas: modo promíscuo e modo não promíscuo. As formas de operação são distintas no seguinte aspecto: em modo não promíscuo só é realizada a captura do tráfego de, para, ou roteado pelo *host* no qual está sendo realizada a captura; por outro lado, em modo promíscuo, é efetuada a captura de todo o tráfego recebido no dispositivo de rede do *host* onde está sendo realizada a captura, ou seja, em um domínio de colisão *ethernet*, por exemplo, é possível capturar todos os pacotes passantes. A Figura 14 ilustra os dois modos de operação da *Libpcap*.

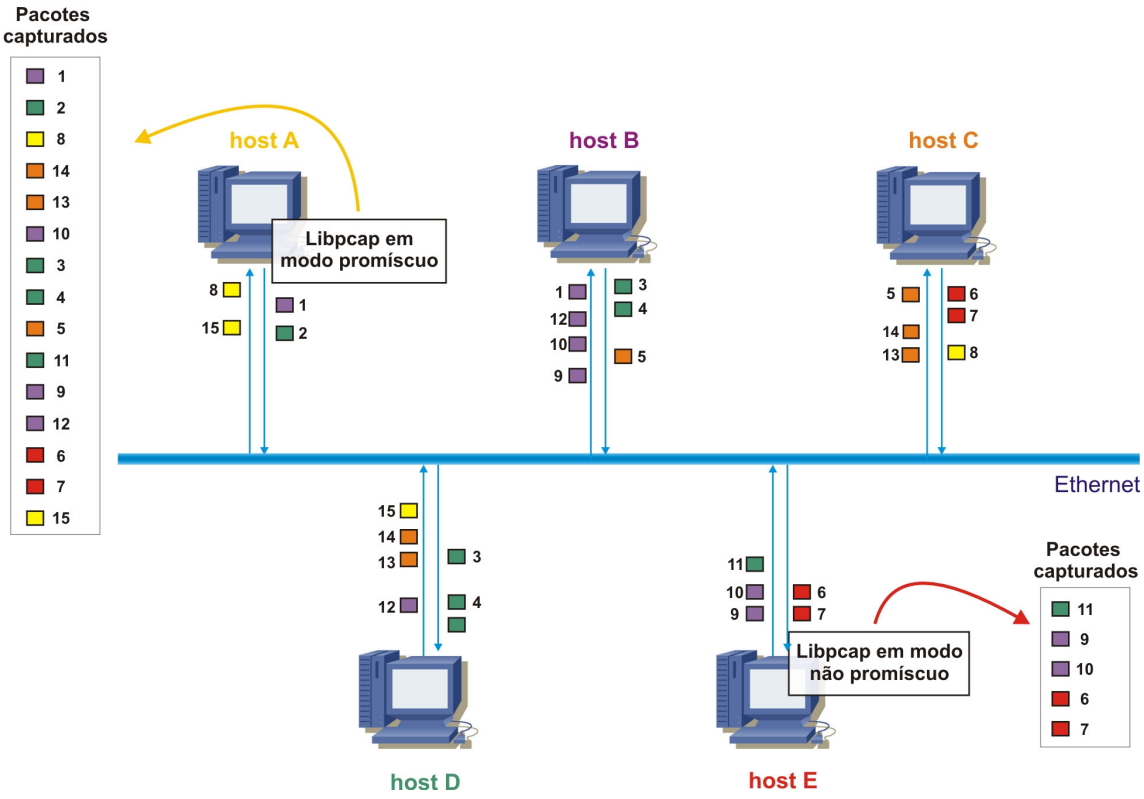



**Figura 14** Ilustração dos dois modos de funcionamento da *Libpcap*.

No presente trabalho, utilizou-se a *Libpcap* em modo promíscuo para que todo o tráfego passante fosse capturado. Além disso, cabe salientar que *Pcap* possui versão portada para o sistema operacional *Windows*, a qual é intitulada como *Winpcap*. As últimas versões da *WinPcap* contam em sua API com dois métodos de amostragem nativos, são eles: *PCAP_SAMP_1_EVERY_N* e *PCAP_SAMP_FIRST_AFTER_N_MS*. Ambos são embasados no algoritmo sistemático de amostragem (ver Figura 6 ), o primeiro baseado em evento e o segundo em tempo. Embora os métodos disponibilizados sejam bastante simplificados, a presença dos mesmos nas versões mais atuais desta biblioteca demonstra o interesse e a demanda por implementações mais sofisticadas de mecanismos de amostragem nas ferramentas de medição.

## 4.2 DESENVOLVIMENTO DO PROTÓTIPO

Nesta seção é apresentado o desenvolvimento propriamente dito da técnica de amostragem aleatória estratificada adaptativa, como módulo adicional ao sistema tradicional de medição de tráfego baseada em fluxos, do GPARC&TI. Para tal, serão enfatizadas as particularidades de implementação da técnica de amostragem, assim como as adaptações realizadas para que as novas funcionalidades estejam em conformidade com os padrões requeridos pelo sistema de medição como um todo.

Inicialmente, é importante ressaltar o cenário global no qual o sistema de medição de tráfego baseado em fluxos está inserido, apresentado na Figura 15 . Resumidamente, todo o processo de medição, incluindo a configuração (Figura 15 - A), captura e análise dos pacotes (Figura 15 - B), identificação dos fluxos (Figura 15 - C) e exportação das informações observadas (Figura 15 - D), está alocado na entidade intitulada "observador IPFIX"[4] (Figura 15 - E).

O transporte das informações coletadas por esta entidade é realizado utilizando o protocolo de comunicação baseado nas recomendações IPFIX (Figura 15 - F), o qual é apresentado por [SAN 07a]. Finalmente, as informações são recebidas pela entidade "Coletor IPFIX" (Figura 15 - G), a qual é responsável por interpretar as mensagens no formato IPFIX (Figura 15 - H) e inseri-las em uma base de dados relacional (Figura 15 - I), para análise posterior (Figura 15 - J).

O desenvolvimento referente ao presente trabalho está alocado em sua totalidade na entidade "observador IPFIX", adaptada para receber parâmetros adicionais e a partir de tais parâmetros realizar a medição do tráfego baseado em fluxos, segundo as premissas da técnica de amostragem aleatória estratificada adaptativa.

---

[4] O conceito "observador IPFIX" é, também, largamente referenciado como "dispositivo IPFIX", generalização utilizada para referenciar esta entidade, pois a mesma pode estar associada a dispositivos que desempenham outras tarefas, como por exemplo, um roteador. Pelo fato do presente trabalho estar focado estritamente no processo de medição, será utilizado, doravante, o termo mais específico "observador IPFIX".

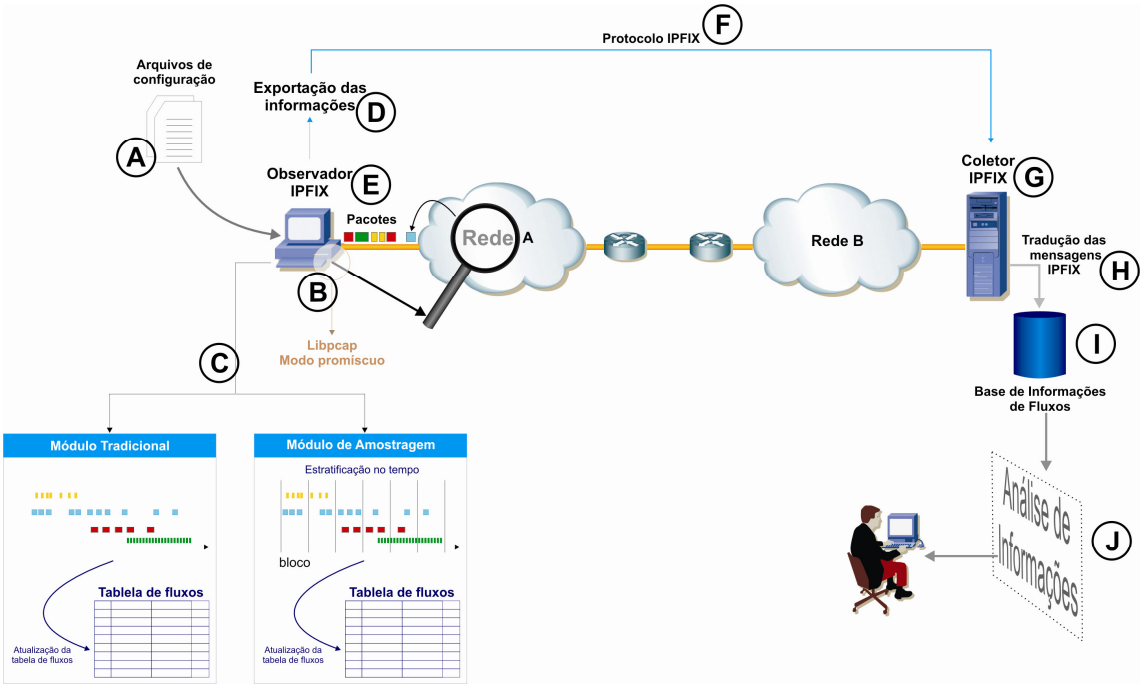



**Figura 15** Cenário global do sistema de medição de tráfego baseado em fluxos.

Partindo deste cenário, as seções seguintes detalham os procedimentos implementados, respectivamente: o processo de caracterização do tráfego em fluxos; os parâmetros que foram adicionados à configuração do sistema e seus respectivos significados, para implementação da técnica de amostragem; como foi realizado o processo de predição utilizando o modelo AR(1) e a seleção dos pacotes a serem amostrados; o desenvolvimento relacionado à divisão do processo de medição em blocos de tempo (estratos), assim como a análise realizada no final de cada bloco; e, finalmente, o processo de exportação utilizando os padrões IPFIX.

### 4.2.1 Caracterização do tráfego baseada em fluxos

Para a caracterização do tráfego em fluxos, baseou-se na abordagem usual, onde um fluxo é determinado por uma tupla composta por cinco atributos: endereço IP de origem, endereço IP de destino, número da porta de origem, número da porta de destino e número de

protocolo[5]. Além disso, para cada fluxo são coletadas as seguintes informações: tempo de duração do fluxo, contabilização do número de pacotes e do volume de *bytes*.

Basicamente, para cada datagrama processado pelo sistema de medição é realizada a extração dos cinco atributos anteriormente citados, os quais são submetidos a uma função *hash*, gerando, desta forma, uma chave identificadora de fluxo. Uma vez gerada a chave identificadora, é realizada a varredura da tabela de fluxos (também referenciada por *cache* de fluxos) para verificar se o fluxo identificado já existe. Em caso afirmativo as informações para o fluxo são atualizadas (contador de pacotes e de *bytes*). Caso contrário uma nova entrada na tabela de fluxos é criada. A ilustração deste processo é apresentada na Figura 16 .

Adicionalmente a este processo, existe um mecanismo para manutenção da tabela de fluxo. Esta necessidade deriva de um outro conceito associado: o *timeout* de expiração dos fluxos. Existe um parâmetro no sistema (*flow_timeout*), no qual é indicado o período máximo de tempo em que o fluxo perdura na tabela de fluxos, desde a sua última atualização. Para tal, cada entrada da tabela de fluxos possui um temporizador associado, que é iniciado toda vez que a entrada é atualizada.

Assim, uma sub-rotina consulta periodicamente a tabela de fluxos e as entradas que expiram o *timeout* especificado, têm suas informações exportadas e, conseqüentemente, são excluídas da tabela. Não existe um consenso quanto um valor ideal para o período de expiração devido à dinâmica dos fluxos, devendo ser feita uma escolha empírica com base nas características da rede em análise.

Desta forma, tem-se a garantia de que a tabela de fluxos será reciclada, fato que possibilita a caracterização distinta de um fluxo com as mesmas propriedades. Por exemplo, se um determinado fluxo foi observado em dado instante do dia (período da manhã) com as seguintes características:

- endereço IP de origem: 192.168.0.15;
- endereço IP de destino: 192.168.0.38;

---

[5] O número identificador de protocolo pertence ao cabeçalho IP e faz referência ao protocolo usado na camada superior no modelo TCP/IP (camada de transporte).

- número da porta de origem: 23800;
- número da porta de destino: 80;
- número de protocolo: 6 (TCP).

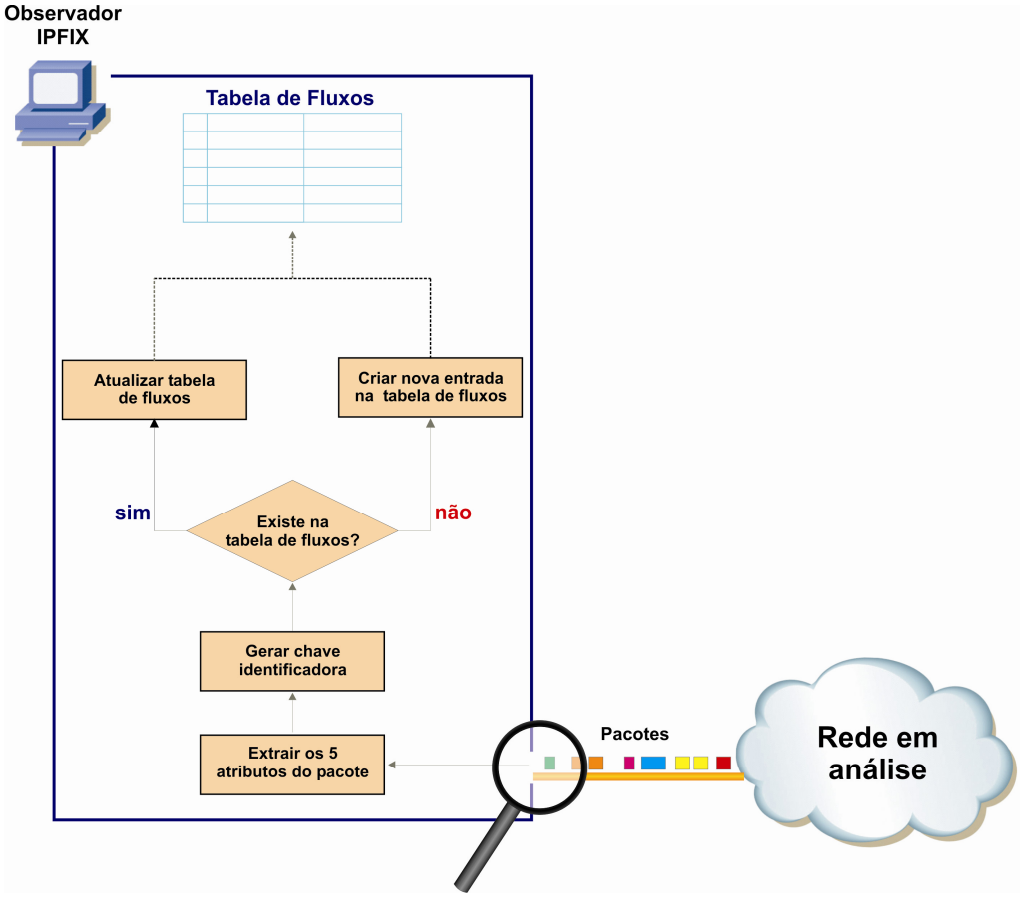



**Figura 16** Processo de identificação de fluxos e atualização da tabela de fluxos.

Considerando que este fluxo seja identificado novamente em um instante posterior (período da tarde), o mesmo será caracterizado como um fluxo totalmente distinto do primeiro, possuindo propriedades diferenciadas, como contabilização de pacotes e contabilização de *bytes*.

Uma outra forma de definir o término de um fluxo pode ser realizada, exclusivamente, para fluxos TCP, baseando-se na *flag FIN* e na *flag RST*, utilizadas para indicar que não existem mais dados a serem transmitidos pelo emissor e *reset* (término) da conexão, respectivamente. O uso deste mecanismo foi implementado no sistema de classificação de

fluxo em adição ao mecanismo de temporizadores. Basicamente, para cada pacote processado é feita a verificação do campo tipo de protocolo da camada superior, no cabeçalho IP. No caso deste campo estar assinalado com o valor “6”, indica que o protocolo usado para o transporte é TCP. Assim, basta verificar se a *flag FIN* ou se a *flag RST* estão assinaladas. A Figura 17 ilustra a adição deste mecanismo ao processo apresentado na Figura 16 .

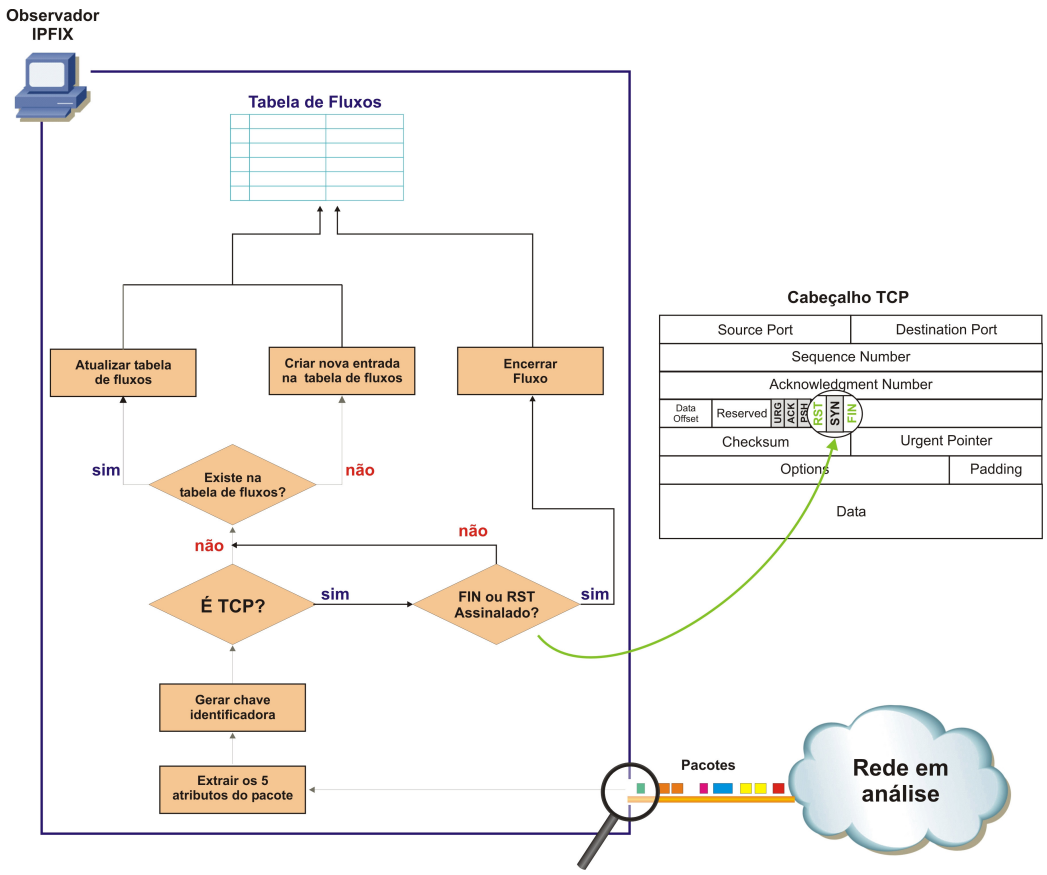



**Figura 17** Encerramento do fluxo a partir da verificação da *flag FIN* e *RST*.

### 4.2.2 Parâmetros de configuração

O arquivo de configuração foi alterado, primeiramente, com o objetivo de permitir flexibilizar o modo de operação do sistema de medição. Especificamente, foi adicionada uma *flag* que indica se o sistema deve utilizar o modelo tradicional para análise de tráfego baseada em fluxos ou alguma técnica de amostragem. Em caso de assinalar o uso de amostragem, deve

ser assinalada a técnica desejada, através de um código pré-definido. Sob esta perspectiva, ao ler o arquivo de configuração o sistema carrega o módulo que deverá utilizar para classificação do tráfego. Atualmente, o sistema suporta apenas a técnica de amostragem aleatória estratificada adaptativa.

Para esta técnica de amostragem, tornou-se necessário adicionar no arquivo de configuração, outros seis parâmetros de inicialização. São eles:

- *block_time*: este parâmetro é um valor inteiro, que representa o intervalo de duração em segundos que cada bloco (estrato) deverá conter;
- *threshold* ( $p^{\theta}$ ): parâmetro que recebe o limiar que indica se um determinado fluxo é ou não elefante. Este parâmetro deve estar no intervalo $0 \leq p^{\theta} \leq 1$;
- *past_predict*: parâmetro que recebe o número de blocos passados a serem utilizados na predição do próximo bloco, pelo modelo AR(1);
- *reliability* ( $\eta$ ): corresponde a confiabilidade das estimativas, segundo um nível de tolerância de erro pré-estabelecido. Este valor deve estar no intervalo $0 \leq \eta \leq 1$;
- *precision* ( $\varepsilon$ ): corresponde a precisão das estimativas, segundo um nível de tolerância de erro pré-estabelecido, por exemplo 0,1;
- *scv* ( $S^{\theta}$ ): corresponde ao quadrado do coeficiente de variação. Especificamente refere-se ao coeficiente que indica a variabilidade do tamanho dos pacotes que compõem um fluxo elefante. Assume valores no intervalo $0{,}00007 \leq S^{\theta} \leq 1$.

Após a leitura dos parâmetros de configuração, ambos são utilizados em seus respectivos contextos. O parâmetro *block_time*, cria um processo auxiliar responsável por gerar um sinal a cada vez que o intervalo de tempo informado é completado. Os parâmetros *past_predict*, *reliability*, *precision* e *scv* são aplicados na expressão (equação (12)) que determina o número mínimo de amostras requeridas para contabilização do número pacotes e do volume de *bytes* para os fluxos caracterizados como “elefantes”. Neste ponto, cabe salientar que tornou-se necessário implementar o inverso da função de distribuição cumulativa ( $\Phi(\cdot)^{-1}$ ). O código-fonte desta implementação encontra-se no ANEXO D.

Finalizando, o parâmetro *past_predict* é utilizado para alocar uma fila *First-In-First-Out* (FIFO) de valores inteiros e capacidade *past_predict + 1*, a qual armazena a quantidade total de pacotes observados nos *past_predict + 1* estratos passados (blocos de tempo).

### 4.2.3 Predição do total de pacotes para o próximo bloco

Conforme apresentado na seção 3.1.3 a probabilidade de amostragem (ver equação (13)) é computada no começo de um novo bloco, através da predição do total de pacotes. Porém, quando o sistema é inicializado não existem valores passados para a predição ser realizada, ou seja, a fila que armazena a contabilização do número de pacotes para os blocos passados não possui as entradas requeridas para a predição.

Para tal, determinou-se um tempo de convergência para o sistema quando da utilização da técnica de amostragem. O tempo de convergência (em segundos) é calculado através do produto do valor informado para o parâmetro *past_predict + 1*, pelo valor informado para o parâmetro *block_time*. A Figura 18 ilustra o fluxograma de funcionamento do processo de convergência.

No encerramento do período de convergência, é realizado o primeiro processo de predição para o próximo bloco e, a partir do número de pacotes estimados, é realizada a seleção aleatória dos pacotes a serem amostrados. A partir daí, o sistema de medição segue o fluxo normal, predizendo o número de pacotes para o próximo bloco no final do bloco corrente e selecionando as futuras amostras. O fluxo geral de funcionamento do sistema de medição utilizando amostragem é apresentado no APÊNDICE B. Para o processo de predição, conforme mencionado anteriormente, foi implementado o modelo AR(1).

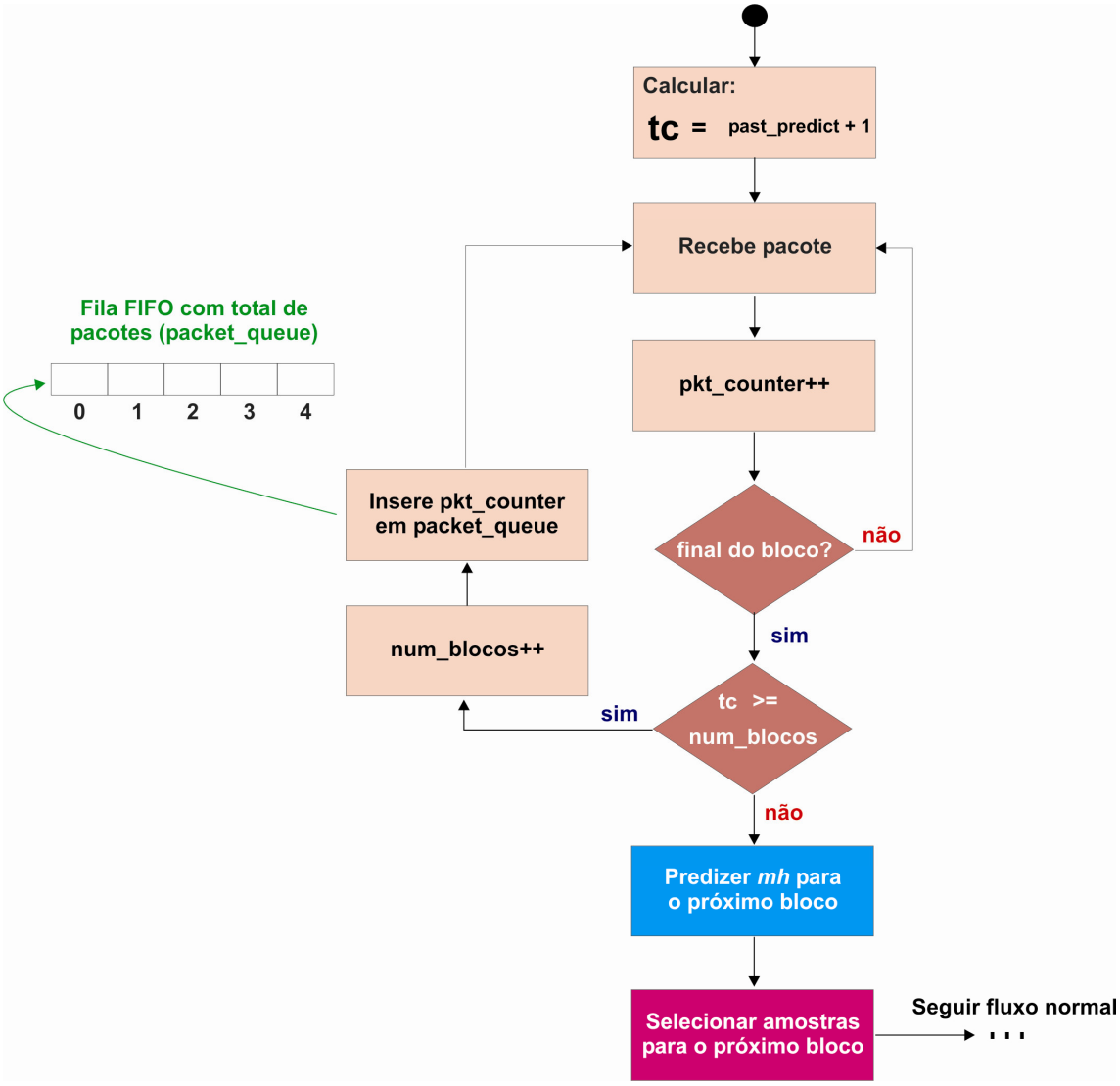



**Figura 18** Fluxograma do processo de convergência.

Na continuação, a seleção das amostras para o bloco seguinte foi implementada utilizando o número de pacotes preditos como base para a definição do espaço amostral. Basicamente, os números aleatórios são sorteados no intervalo $[1; m_h]$, onde $m_h$ é o número de pacotes estimados pelo sistema de predição. Supondo-se, por exemplo, que o número de amostras é $n^* = 5$ (constante conforme a equação(12)) e a predição para o total de pacotes do próximo bloco é $m_h = 100$, aloca-se uma lista associativa $L[1...m_h]$ (no C++ chamada de *map*) de $n^*$ posições, que possui em seus índices os números aleatórios sorteados no intervalo entre [1; 100 ($m_h$)] e, para cada índice, o conteúdo "*true*".

A utilização deste mecanismo torna bastante simplificado o processo de implementação, pois basta manter um contador de pacotes e para cada novo pacote recebido o contador é utilizado para referenciar uma posição na lista. Caso o conteúdo retornado pelo índice consultado seja verdadeiro o pacote é amostrado, caso contrário o pacote é descartado. A Figura 19 ilustra este processo. A implementação do modelo AR(1), juntamente com o processo de seleção das amostras pode ser visualizado no APÊNDICE D.

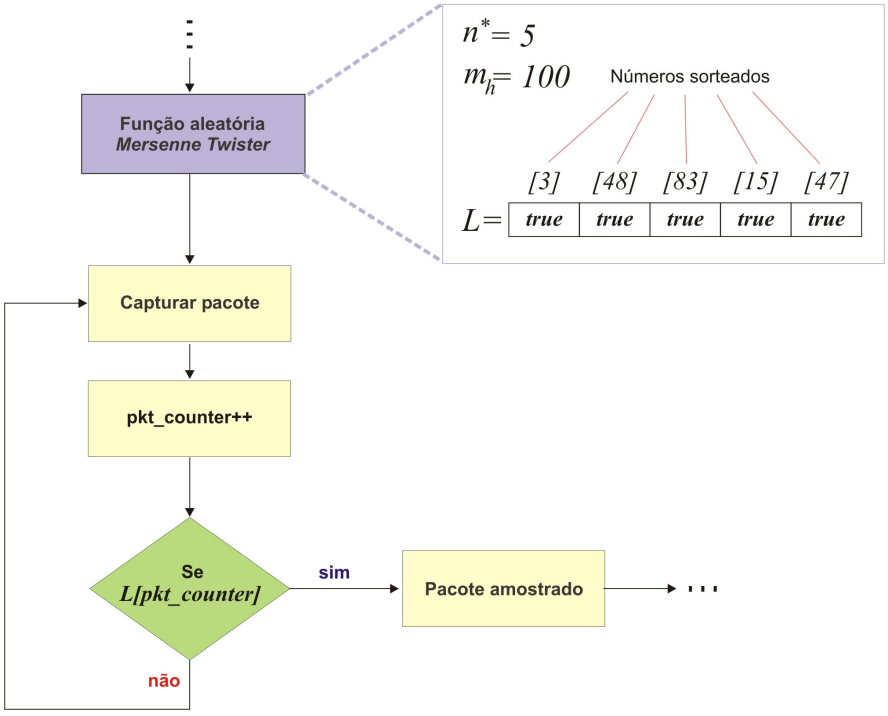



**Figura 19** Processo de seleção dos pacotes amostrados.

### 4.2.4 Divisão e processamento dos blocos

Conforme mencionado, a partir do valor informado para configuração do parâmetro *block_time* é criado um processo auxiliar, que se torna responsável por gerar um sinal a cada vez que um bloco de tempo (estrato) é completado. Segundo esta abordagem, torna-se necessário criar um mecanismo diferenciado para a classificação do tráfego em fluxos, assim como para contabilização do total de pacotes e do volume de *bytes*.

Esta tomada de decisão é necessária, pois a aplicação da técnica de amostragem não pode interferir na estrutura do sistema de medição original. Resumidamente, o sistema utilizando a abordagem tradicional captura os pacotes, classifica-os, atualiza a tabela de fluxos e exporta as informações. Estes procedimentos devem ser mantidos, porém a divisão em blocos de tempo, as rotinas agregadas para cálculo das estimativas e a classificação do fluxo como "elefante", determinam a implementação de procedimentos adicionais, aplicados apenas no uso da técnica de amostragem.

Para apresentar claramente os procedimentos adicionais implementados, assim como as pequenas adaptações realizadas no corpo do *software* é relevante destacar, inicialmente, quais as informações que compõem a tabela de fluxos. Um resumo de tais informações e seus respectivos significados está apresentado na Tabela 6 .

**Tabela 6** Resumo dos campos da tabela de fluxos.

| Campo | Descrição |
|---|---|
| *flowIdentifier* | Chave identificadora do fluxo, a qual é composta pelos atributos apresentados na seção 4.2.1 |
| *flowDbIdentifier* | Chave identificadora do fluxo na base de dados, a qual é composta pelos atributos apresentados na seção 4.2.1 e, adicionalmente, o *timestamp* do primeiro pacote observado para fluxo |
| *flowStartSeconds* | *Timestamp* em que o fluxo foi observado |
| *packetTotalCountIn* | Total de pacotes contabilizados para fluxo no sentido *upstream* |
| *packetTotalCountOut* | Total de pacotes contabilizados para fluxo no sentido *downstream* |
| *octetTotalCountIn* | Total de bytes contabilizados para fluxo no sentido *upstream* |
| *octetTotalCountOut* | Total de bytes contabilizados para fluxo no sentido *downstream* |

Com base na tabela de informações de fluxo, a primeira alteração que se tornou necessária foi à adição de um novo campo. Este campo adicional foi incluído para contabilizar o total de pacotes observados pelos blocos em que o fluxo perdurou. Assim, quando o fluxo expira a verificação que determina se o mesmo é ou não "elefante" pode ser realizada diretamente.

Entretanto, a principal funcionalidade adicionada é à classificação do tráfego interna a cada bloco de tempo. Pela forma com que o método de amostragem foi concebido, as estimativas das informações dos fluxos são realizadas no final de cada bloco, ou seja, o real preenchimento da tabela de fluxos deve ser realizado, analogamente, no final de cada estrato.

Por outro lado, o sistema de medição tradicional atualiza a tabela de fluxos a cada novo pacote observado. Por este motivo, identificou-se uma particularidade que poderia comprometer a interoperabilidade do módulo que implementa a técnica de amostragem com o núcleo do sistema de medição tradicional. Para solucionar este impasse, utilizou-se um mecanismo de tabela de fluxos temporária.

Basicamente, no decorrer de cada bloco é criada uma tabela temporária de fluxos a partir dos pacotes amostrados. O preenchimento desta tabela segue os mesmo critérios do sistema de medição tradicional. Assim, no final do bloco têm-se as informações desejadas: o total de pacotes observados no bloco (total de pacotes independente de fluxos) e a classificação do tráfego em fluxos com base nos pacotes amostrados. Com posse dessas informações a tabela temporária de fluxos é atualizada, recebendo o valor das estimativas calculadas para a contabilização de pacotes e do volume de *bytes* para cada fluxo (equação (14)). Após o encerramento do processo de atualização da tabela temporária de fluxos, cada entrada que compõe esta tabela passa a ser atualizada na tabela global de fluxos atualizando também o campo adicional que contém o somatório do total de pacotes observados nos blocos pelos quais o fluxo perdurou.

Na rotina de migração das informações contidas na tabela temporária de fluxos para a tabela global de fluxos utilizam-se os mesmos critérios. Ou seja, caso um novo fluxo seja observado no interior do bloco, esta entrada também irá se consolidar como uma nova entrada na tabela global de fluxos. Por outro lado, para o caso de fluxos anteriormente observados, a tabela global já terá uma entrada contendo a mesma chave identificadora, bastando atualizar as informações. A Figura 20 apresenta um exemplo do mecanismo de tabelas temporárias de fluxo utilizando um número reduzido de campos para as tabelas de fluxos (temporária e global), apenas com o objetivo de ilustrar o procedimento.

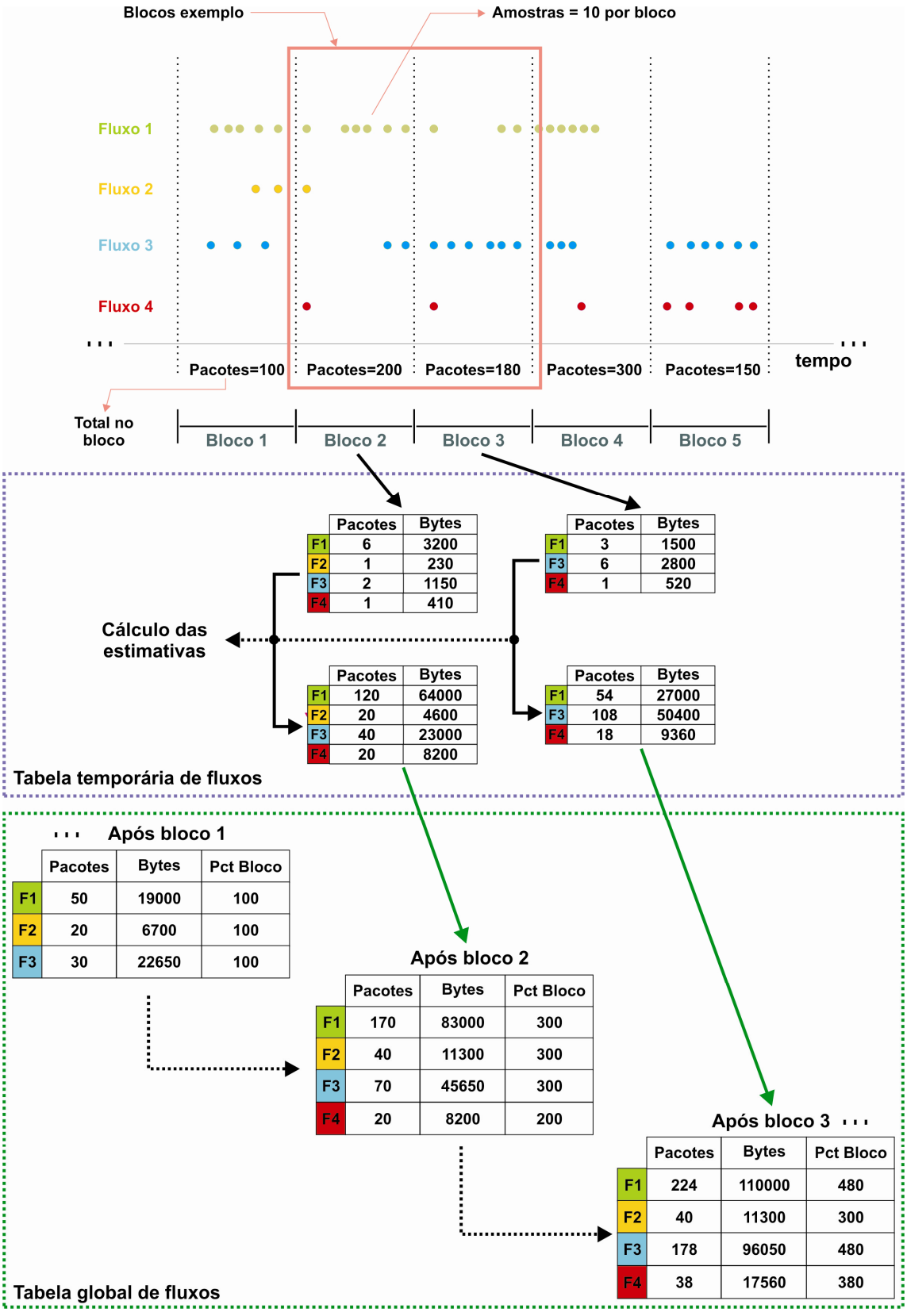



**Figura 20** Mecanismo de tabelas temporárias de fluxo.

### 4.2.5 Processo de exportação das informações de fluxo

Conforme mencionado anteriormente, para o processo de exportação das informações de fluxo utilizou-se o sistema proposto em [SAN 07a]. Seguindo as recomendações do IPFIX, foram implementados dois critérios para exportação dos fluxos: o primeiro, segue o procedimento explicitado na seção 4.2.1, no qual um fluxo é exportado toda vez em que o seu *timeout* é expirado; o segundo critério caracteriza-se no contexto de fluxos de longa duração, para os quais a exportação é realizada de forma periódica.

Para o segundo critério, especificamente, utilizou-se um parâmetro adicional no arquivo de configuração, onde é informado um intervalo de tempo em segundos indicando a periodicidade de exportação (também pode ser utilizado para o período de exportação o valor do campo *reportPeriod* da relação "*observer*" na base de dados, APÊNDICE C). Com relação às informações exportadas para cada fluxo, utilizou-se um *template* IPFIX composto pelos campos detalhados na Tabela 7 .

A especificação de tais campos é realizada na configuração do sistema de medição, através de um documento XML. Analogamente, estas informações compõem a base de dados que recebe as informações exportadas. O modelo Entidade-Relacionamento (ER) desta base de dados é apresentado no APÊNDICE C.

**Tabela 7** Informações exportadas pelo sistema de medição.

| Campo | Descrição |
|---|---|
| *flowIdentifier* | Chave identificadora do fluxo, a qual é composta pelos atributos apresentados na seção 4.2.1. |
| *flowDbIdentifier* | Chave identificadora do fluxo na base de dados, a qual é composta pelos atributos apresentados na seção 4.2.1 e, adicionalmente, o *timestamp* do primeiro pacote observado para fluxo. Utilizado para manter integridade no caso para fluxos de longa duração, que têm suas informações exportadas periodicamente. |
| *flowStartSeconds* | *Timestamp* em que o fluxo foi detectado. É considerado para o primeiro pacote observado no fluxo. |
| *sourceTransportPort* | Número da porta no transmissor. |
| *sourceIPv4Address* | Endereço IP do transmissor. |
| *destinationTransportPort* | Número da porta no receptor. |
| *destinationIPv4Address* | Endereço IP do receptor. |
| *protocolIdentifier* | Número identificador do protocolo na camada superior. |
| *octetDeltaCountIn* | Volume de octetos observados no sentido *downstream*, desde a última exportação. |
| *octetDeltaCountOut* | Volume de octetos observados no sentido *upstream*, desde a última exportação. |
| *octetTotalCountIn* | Total de octetos contabilizados no sentido *downstream*. |
| *octetTotalCountOut* | Total de octetos contabilizados no sentido *upstream*. |
| *packetDeltaCountIn* | Número de pacotes observados no sentido *downstream*, desde a última exportação. |
| *packetDeltaCountOut* | Volume de pacotes observados no sentido *upstream*, desde a última exportação. |
| *packetTotalCountIn* | Total de pacotes contabilizados no sentido *downstream*. |
| *packetTotalCountOut* | Total de pacotes contabilizados no sentido *upstream*. |

## 4.3 AMBIENTES DE TESTE

A presente seção tem por objetivo apresentar o ambiente de teste utilizado para avaliação dos resultados obtidos a partir do protótipo desenvolvido. Para tal, optou-se por utilizar um cenário de rede controlado de menor complexidade física e lógica, objetivando minimizar o número de variáveis envolvidas no processo de avaliação, gerando, conseqüentemente, facilidades na interpretação dos resultados.

Esta tomada de decisão tem como base o fato de que a avaliação do protótipo deve estar inserida em um contexto com tráfego relativamente alto, situação que pode ser representada com o cenário de rede utilizado. Além disso, o principal objetivo da avaliação está centrado na verificação do grau de confiabilidade das informações obtidas com o uso da técnica de amostragem, fato que, novamente, não exige uma estrutura de rede complexa.

Cabe ressaltar que, para realização dos testes, utilizou-se uma metodologia onde devem ser destacados:

- cenário de rede: a topologia lógica da rede, na qual foram realizados os testes;
- condições de contorno e operação: apresenta as especificidades dos dispositivos envolvidos nos testes, tais como: fabricante, *hardware*, SO, entre outras;
- metodologia empregada: a metodologia utilizada para o desenvolvimento dos testes;
- resultados esperados: referente aos procedimentos de teste propostos e apresentação os resultados esperados.

Os resultados obtidos, bem como as discussões a cerca dos resultados, são apresentados no capítulo 5 .

### 4.3.1 Modelo do cenário de rede

Nesta seção é detalhado o modelo do cenário de rede utilizado nos testes. A Figura 21 ilustra a estrutura lógica da topologia de rede utilizada. Basicamente, utilizou-se um ambiente com capacidade de 100 *Mbps* operando em modo *full-duplex*, onde o tráfego gerado para análise trafega nos dois sentidos (*upload* e *download*), entre o *host* A e o *host* B. O detalhamento das informações relacionadas às características dos dispositivos de rede e *host´s* será apresentado na próxima seção (4.3.2), referente a condições de contorno e operação.

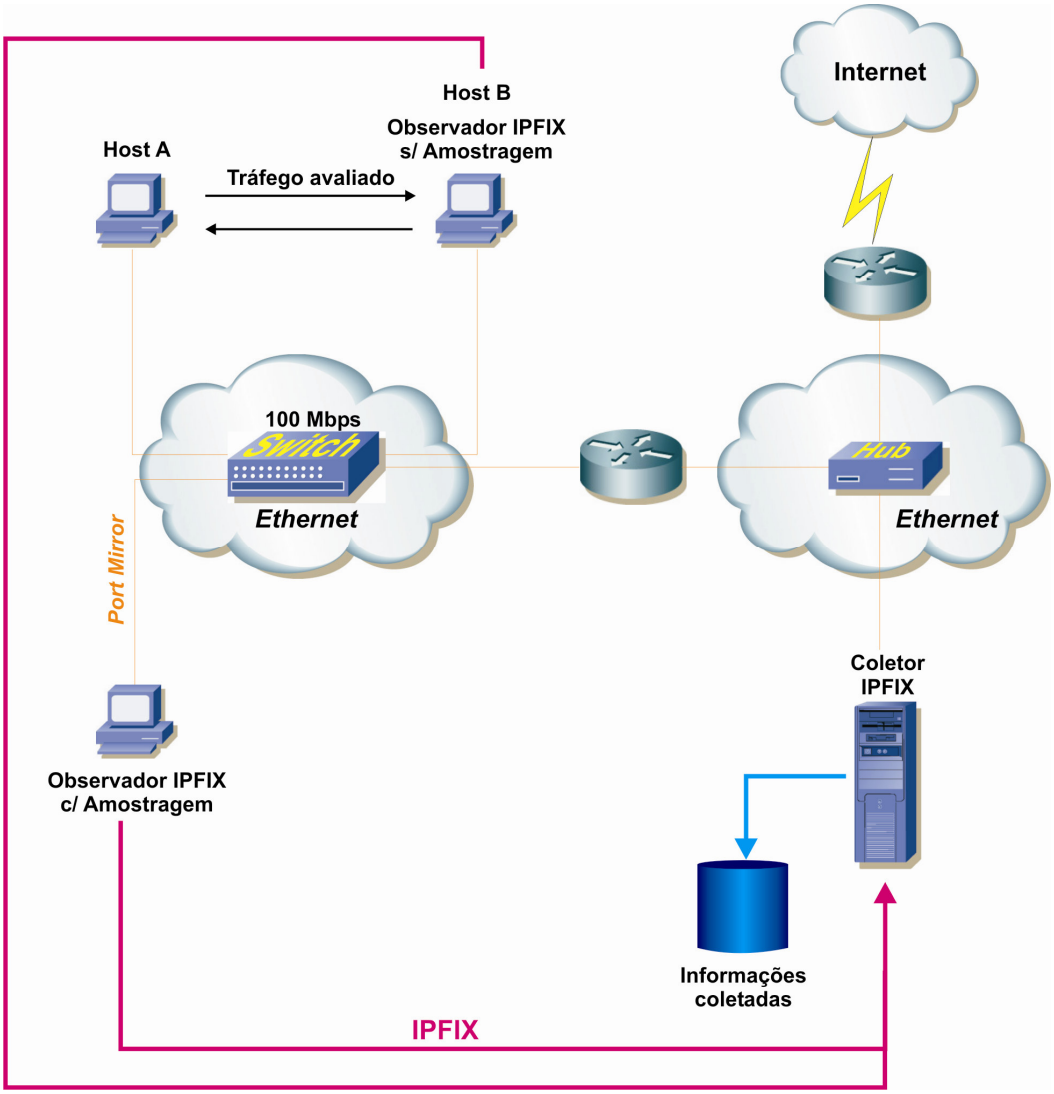



**Figura 21** Cenário de rede utilizado nos testes.

### 4.3.2 Condições de contorno e operação

Abaixo são apresentadas as informações relacionadas aos dispositivos de rede e *host´s* utilizados na realização dos testes, segundo o ambiente de rede mostrado na Figura 21 .

- *Host´s* e "observador IPFIX": Estes equipamentos possuem as mesmas características de *hardware* e *software*, conforme apresentado na Tabela 8 .

**Tabela 8** *Host´s* e "observador IPFIX".

| | |
|---|---|
| **Processador** | *Pentium* 4 HT – 3.00 *GHz* |
| **Memória** | 1.00 *GByte* |
| **Interface de Rede** | *Fast Ethernet* 100 *Mbps* |
| **Observações** | Operam a 100 *Mbps full-duplex*. |
| **Sistema Operacional** | *Debian Linux* – Kernel 2.6.12.1 |

- Coletor IPFIX: Possui as características de *hardware* e *software* apresentadas na Tabela 9 .

**Tabela 9** Coletor IPFIX.

| | |
|---|---|
| **Processador** | *Intel(R) Xeon(TM)* - 3.00 *GHz* |
| **Memória** | 2.00 *GBytes* |
| **Interface de Rede** | *Gigabit Ethernet* 1 *Gbps* |
| **Observações** | Por estar conectado ao *hub*, opera a 10 *Mbps half-duplex*. |
| **Sistema Operacional** | *Debian Linux* - Kernel 2.6.12.1 i686 |

- *Hub*: Possui as características apresentadas na Tabela 10 .

**Tabela 10** Características do *hub*.

| | |
|---|---|
| **Modelo** | ESH – 709 / 10 *Mbps* |
| **Fabricante** | Encore |
| **Número de portas** | 8 portas (*Ethernet*) |
| **Observações** | Este *hub* opera em modo *half-duplex* compartilhado, com capacidade nominal de 10 *Mbps*. |

- *Switch*: Possui as características apresentadas na Tabela 11 .

**Tabela 11** Características do *Switch*.

| | |
|---|---|
| **Modelo** | *Switch* 3250 *SuperStack* 3 |
| **Fabricante** | *3Com* |
| **Número de portas** | 48 portas de 100 *Mbps* e 2 portas de 1 *Gbps* (*Ethernet*) |
| **Observações** | Este *switch* opera em modo *full-duplex*, atingindo desta forma a capacidade de 100 *Mbps* em ambos os sentidos de transmissão. |

Os roteadores que compõem a estrutura de rede apresentada, não são representativos nos testes, por este motivo não serão detalhados. O roteador que interliga a rede onde estão

alocados os *host´s* e os "observadores IPFIX" com a rede na qual está instalado o coletor IPFIX, apenas representa uma abstração para ilustrar a conectividade entre duas redes distintas. Da mesma forma, o outro roteador foi adicionado a topologia para abstrair a complexidade que envolve a conexão da rede utilizada nos testes com a *Internet*.

Cabe salientar que para montar o cenário de rede idealizado, utilizou-se uma *feature* disponibilizada pelo *Switch*. Pelo fato deste equipamento ter sido configurado em modo *full-duplex* para uma rede micro-segmentada, tornou-se necessário gerar tráfego de interesse entre dois *hosts* e efetuar o espelhamento de portas. No modelo de *Switch* utilizado, só é possível espelhar portas na relação 1:1, ou seja, não se tem a possibilidade de fazer o espelhamento do tráfego de *N* portas para uma porta específica.

Com relação aos *softwares* aplicados nos testes utilizou-se, basicamente, um aplicativo para transferência de arquivos sob o protocolo *Hypertext Transfer Protocol* (HTTP), intitulado *wget*. Este aplicativo utiliza para transporte dos dados o protocolo TCP. Adicionalmente, utilizou-se um aplicativo desenvolvido no próprio GPARC&TI, para a geração de tráfego UDP, o *udpflood*. Este aplicativo possibilita determinar as características do tráfego UDP a ser injetado na rede, através dos seguintes parâmetros: número de pacotes, tamanho dos pacotes, intervalo de transmissão de cada pacote, endereço de destino, porta de destino e porta de origem.

### 4.3.3 Método empregado

De acordo com o cenário de rede apresentado, foram realizados diferentes testes sempre objetivando averiguar as divergências oriundas dos resultados obtidos com o sistema de medição tradicional e o sistema utilizando o módulo de amostragem. Esta característica foi determinante para a tomada de decisão de utilizar dois pontos de observação, pois desta forma conseguiu-se garantir que ambos os observadores efetuassem a captura das mesmas informações, em condições análogas.

Partindo dessa premissa, o cenário elaborado visa submeter o sistema de medição a um ambiente no qual se consegue atingir uma capacidade de transmissão relativamente alta (100 *Mbps*). Para tal, foram realizados dois conjuntos de procedimentos de teste. O primeiro conjunto foi elaborado com o objetivo de observar o comportamento das estimativas utilizando apenas fluxos de interesse, transmitidos de forma paralela não-sincronizada e sem a existência de tráfego competitivo. A não existência de tráfego competitivo teve como objetivo facilitar o controle das estimativas na alternância dos coeficientes de precisão, confiabilidade e tamanho dos estratos. A Tabela 12 apresenta os parâmetros utilizados em cada um dos procedimentos do primeiro conjunto de teste.

**Tabela 12** Parâmetros utilizados para realização do primeiro conjunto de testes.

| | $\eta$ | $\varepsilon$ | $P^{\theta}$ | $S^{\theta}$ | $n^{*,b}$ | *past_predict* | Bloco (segundos) |
|---|---|---|---|---|---|---|---|
| Procedimento 1 | 0.1 | 0.1 | 0.01 | 0.2 | 32196 | 5 | 10 |
| Procedimento 2 | 0.1 | 0.1 | 0.01 | 0.2 | 32196 | 5 | 20 |
| Procedimento 3 | 0.1 | 0.1 | 0.01 | 0.2 | 32196 | 5 | 30 |
| Procedimento 4 | 0.2 | 0.2 | 0.01 | 0.2 | 4886 | 5 | 10 |
| Procedimento 5 | 0.2 | 0.2 | 0.01 | 0.2 | 4886 | 5 | 20 |
| Procedimento 6 | 0.2 | 0.2 | 0.01 | 0.2 | 4886 | 5 | 30 |
| Procedimento 7 | 0.3 | 0.3 | 0.01 | 0.2 | 1420 | 5 | 10 |
| Procedimento 8 | 0.3 | 0.3 | 0.01 | 0.2 | 1420 | 5 | 20 |
| Procedimento 9 | 0.3 | 0.3 | 0.01 | 0.2 | 1420 | 5 | 30 |

Para a transmissão dos fluxos de interesse, foi utilizada a transferência de arquivos sob o protocolo HTTP, através do aplicativo *wget*, conforme citado anteriormente. Os arquivos utilizados foram criados artificialmente e com tamanhos de interesse. Para criação dos arquivos utilizou-se uma rotina bastante simples disponibilizada no SO *Linux*: "*dd if=/dev/zero of=[arquivo destino] bs=[tamanho do bloco] count=[numero_de_blocos]*". As características dos arquivos utilizados nos procedimentos do primeiro conjunto de teste são apresentadas na Tabela 13 .

**Tabela 13** Arquivos utilizados no primeiro conjunto de testes.

| | Tamanho |
|---|---|
| Arquivo 1 | 1 *GBytes* (1073741824 octetos) |
| Arquivo 2 | 800 *Mbytes* (838860800 octetos) |
| Arquivo 3 | 500 *Mbytes* (524288000 octetos) |
| Arquivo 4 | 100 *Mbytes* (104857600 octetos) |
| Arquivo 5 | 50 *Mbytes* (52428800 octetos) |
| Arquivo 6 | 10 *Mbytes* (10485760 octetos) |
| Arquivo 7 | 1 *Mbytes* (1048576 octetos) |
| Arquivo 8 | 500 *Kbytes* (512000 octetos) |

O segundo conjunto de procedimentos de teste foi concebido com o intuito de averiguar os resultados das estimativas na existência de tráfego competitivo. Além disso, foi realizada uma pequena variação nas características dos fluxos de interesse, apenas com o intuito de gerar um ambiente com fluxos menores. O tráfego competitivo foi injetado no cenário de teste utilizando o aplicativo *udpflood*, com as configurações apresentadas abaixo:

- número de pacotes: 10000000;
- tamanho dos pacotes: 800 *bytes*;
- intervalo entre os pacotes: 0 segundos.

Para os fluxos de interesse utilizou-se o mesmo método empregado para o primeiro conjunto de procedimentos de teste. A Tabela 14 apresenta os parâmetros utilizados em cada um dos procedimentos do segundo conjunto de teste. A 0apresenta as características dos arquivos utilizados para geração dos fluxos de interesse.

**Tabela 14** Parâmetros utilizados para realização do segundo conjunto de testes.

| | $\eta$ | $\varepsilon$ | $P^{\theta}$ | $S^{\theta}$ | $n^{*,b}$ | *past_predict* | Bloco (segundos) |
|---|---|---|---|---|---|---|---|
| Procedimento 10 | 0.1 | 0.1 | 0.01 | 0.2 | 32196 | 5 | 10 |
| Procedimento 11 | 0.2 | 0.2 | 0.01 | 0.2 | 4886 | 5 | 10 |
| Procedimento 12 | 0.3 | 0.3 | 0.01 | 0.2 | 1420 | 5 | 10 |

**Tabela 15** Arquivos utilizados no segundo conjunto de testes.

| | Tamanho |
|---|---|
| Arquivo 1 | 1 *GBytes* (1073741824 octetos) |
| Arquivo 2 | 800 *Mbytes* (838860800 octetos) |
| Arquivo 3 | 500 *Mbytes* (524288000 octetos) |
| Arquivo 4 | 100 *Mbytes* (104857600 octetos) |
| Arquivo 5 | 500 *Kbytes* (512000 octetos) |
| Arquivo 6 | 100 *Kbytes* (102400 octetos) |
| Arquivo 7 | 5 *Kbytes* (5120 octetos) |

Em ambos os procedimentos de teste o período de exportação, utilizado para os fluxos de longa duração, é igual ao tamanho dos blocos. Cabe ressaltar que, para o sistema tradicional de medição, utilizou-se os mesmos valores para o intervalo de exportação. Os resultados gerados pelas medições foram, em todos os testes realizados, direcionados para armazenamento em banco de dados, facilitando assim a posterior análise dos dados.

Outro procedimento de teste realizado refere-se à análise do sistema de predição do total de pacotes para o próximo bloco. Basicamente utilizou-se um cenário análogo ao empregado no primeiro procedimento de teste do primeiro conjunto, variando apenas o número de blocos passados utilizados na predição e a forma de transmissão dos fluxos de interesse, a qual foi realizada de forma a gerar uma oscilação significativa no tráfego. A Tabela 16 apresenta os parâmetros utilizados para esta avaliação.

**Tabela 16** Parâmetros utilizados para do modelo AR(1).

| | $\eta$ | $\varepsilon$ | $P^{\theta}$ | $S^{\theta}$ | $n^{*,b}$ | *past_predict* | Bloco (segundos) |
|---|---|---|---|---|---|---|---|
| Procedimento 1 | 0.1 | 0.1 | 0.01 | 0.2 | 32196 | 3 | 10 |
| Procedimento 2 | 0.1 | 0.1 | 0.01 | 0.2 | 32196 | 5 | 10 |
| Procedimento 3 | 0.1 | 0.1 | 0.01 | 0.2 | 32196 | 7 | 10 |

Os resultados obtidos e as discussões a cerca dos resultados, serão apresentadas no capítulo seguinte (capítulo 5 ).

### 4.3.4 Resultados esperados

Para o primeiro conjunto de testes realizados é esperado que a ferramenta de medição utilizando a técnica de amostragem implementada apresente degradação nas estimativas com o aumento dos valores atribuídos para os coeficientes de confiabilidade e precisão, assim como no aumento do tamanho dos estratos. Para o segundo conjunto de testes, a expectativa é bastante semelhante, sendo que a perda nas estimativas deverá estar associada, obviamente, a diminuição do número mínimo de amostras.

Com relação à avaliação realizada para o modelo AR(1), é esperado que as oscilações no tráfego sejam identificadas em todos os procedimentos, sendo que o objeto principal de análise é averiguar qual dos parâmetros utilizados apresentou melhor adaptação, convergindo mais rapidamente para o resultado real.

É importante ressaltar que os resultados obtidos para os fluxos de interesse, deverão sofrer uma pequena distorção (até mesmo no sistema de medição tradicional), devido à contabilização de cabeçalhos, uma vez que ambos as abordagens consideram o *overhead*. Outro ponto a ser destacado, refere-se à questão da contabilização ser considerada, especificamente nos testes realizados, apenas no sentido de *downstream*, subtraindo, desta forma, o tráfego agregado proveniente do controle de conexão inerente ao protocolo TCP. Esta abordagem visa, simplesmente, facilitar a interpretação dos resultados.

# 5 RESULTADOS E DISCUSSÕES

O presente capítulo tem por objetivo apresentar os resultados obtidos e as discussões relacionadas ao desenvolvimento do trabalho proposto. Inicialmente, são apresentadas discussões relacionadas a forma como o protótipo foi implementado, indicando as principais particularidades. No decorrer do capítulo, serão apresentados os resultados obtidos a partir dos testes realizados com a técnica de amostragem implementada, destacando, especificamente: constatações referentes a predição do total de pacotes para o próximo bloco, e o comparativo entre os resultados reais e as estimativas para o número de pacotes e o volume de *bytes*.

## 5.1 DESENVOLVIMENTO DO SISTEMA DE MEDIÇÃO

Como principal resultado obtido no protótipo desenvolvido, pode-se destacar a interoperabilidade mantida com o sistema tradicional de medição de tráfego baseada em fluxos. A implementação da técnica de amostragem como um módulo adicional, possibilitou facilidades não só para legibilidade e entendimento do núcleo do *software*, mas também para expansibilidade. Além disso, em ambientes onde não for desejável o uso de amostragem, o sistema de medição tradicional se mantém inalterado, sendo necessário apenas o ajuste nos parâmetros de configuração.

O sistema de exportação das estatísticas de fluxo utilizando a adaptação da biblioteca *libIPFIX*, desenvolvida por [SAN 07a], também denotou outro ponto de interesse no sistema implementado. A conformidade com os padrões recomendados e documentados no IETF (no caso o IPFIX), possibilita manter, neste contexto o sistema como um todo, alinhado as tendências que passarão a ser exigências efetivas de mercado. Adicionalmente, o uso desta API representou facilidades no processo de exportação das informações de fluxo, além de instrumentalizar o armazenamento destas informações em um SGBD, prática que vem sendo largamente usada nos principais sistemas de gerenciamento de redes.

Por outro lado, cabe ressaltar que embora o uso de um meio de armazenamento robusto como o SGBD *PostGreSQL*, concatenado ao uso de amostragem de pacotes, não elimina a necessidade de customizações nas estruturas de armazenamento das estatísticas de fluxo, principalmente, para ganho de desempenho. Em fluxos de grande duração, existe a necessidade de exportação das informações com determinada periodicidade, conforme apresentado na seção 4.2.5.

Neste caso, especificamente em ambientes onde o perfil de utilização da rede seja predominantemente grande, ou seja, ambientes hostis nos quais os usuários tenham capacidade de largura de banda e liberdade para consumir recursos, deve-se estar atento no momento de definir o modo de armazenamento das informações. Embora o diagrama ER apresentado no APÊNDICE C esteja normalizado enquanto modelo abstrato, o mesmo pode sofrer alterações substanciais no momento de sua implementação, principalmente quando submetido a contextos de rede como o descrito acima.

Para estas situações, vislumbra-se a aplicação de mecanismos pertinentes de rotatividade das informações contidas no banco de dados, como garantia de diminuição na carga de processamento das transações de acesso. Outro ponto a ser observado é a criação de índices em campos-chave. Para o modelo ER apresentado, uma boa aproximação seria a criação de índice no campo "*flowIdentifier*", presente nas relações *flow* e *flowEntry*. A aplicação do mecanismo de visões, também representa um meio para ganho de desempenho, uma vez que as visões são mantidas em memória.

A verificação das *flags FIN* e *RST*, para fluxos TCP, representa outra *feature* adicionada ao sistema, a qual é pertinente independentemente do uso de amostragem ou não. Em sua implementação original, o sistema tradicional de medição de tráfego baseado em fluxos, não efetuava a verificação destes campos do cabeçalho TCP. Com a adição desta rotina, consegue-se diminuir significativamente a permanência desnecessária de entradas na tabela de fluxos, além de subtrair o número de temporizadores e verificações de expiração de *timeout*.

No contexto de implementação do modelo AR(1), averiguou-se que a utilização de uma fila do tipo FIFO para armazenamento da quantidade total de pacotes observados nos blocos passados é bastante adequada, pois ao término de cada bloco a contabilização do total de pacotes deve ser atualizada e a informação mais antiga excluída. Desta forma, a simples

inclusão do novo valor garante a exclusão automática do valor mais antigo, proporcionando facilidades diretas no desenvolvimento do algoritmo, assim como na reciclagem dos valores observados. A avaliação do comportamento do modelo AR(1), quanto a precisão dos valores preditos será apresentada na próxima seção 5.2.1.

Ainda no contexto relacionado ao ajuste da taxa de amostragem para cada estrato a partir do valor predito pelo modelo AR(1), observou-se algumas particularidades de implementação para a seleção dos números aleatórios. Conforme apresentado na seção 4.2.3, os números aleatórios sorteados no intervalo $[1; m_h]$ são armazenados em uma estrutura de lista associativa, a qual contém em seus índices os valores aleatórios sorteados e em seu conteúdo o valor *boolean true*.

A utilização deste mecanismo possibilitou transpassar um problema relacionado à repetição no sorteio de valores aleatórios, dentre o espaço amostral. Especificamente, o uso desta estrutura de dados evita diretamente que não sejam armazenadas entradas duplicadas, uma vez que a mesma possui restrições nativas quanto à inserção de chaves repetidas. Entretanto, observou-se um comportamento indesejável: utilizando-se um algoritmo que realiza um número de iterações igual ao número de amostras requeridas, tem-se uma perda significativa na quantidade de valores aleatórios obtidos.

Por exemplo, partindo dos seguintes parâmetros para a equação (12): $\{\eta, \varepsilon\} = \{0{,}1;0{,}1\}$ e $\{P^{\theta}, S^{\theta}\} = \{0{,}01;0{,}2\}$, têm-se um número mínimo de amostras requeridas igual a $n^{*,b} = 32196$. Para este caso, utilizando-se a aproximação de efetuar 32196 iterações sorteando números aleatórios no intervalo entre $[1; m_h]$, identificou-se perda na quantidade de valores aleatórios sorteados pela incidência de valores repetidos, principalmente quando o valor de $m_h$ é mais próximo de $n^{*,b}$.

Para ilustrar este comportamento, a Figura 22 exibe um gráfico que representa a quantidade de valores aleatórios sorteados sem repetição, utilizando o número de iterações igual a $n^{*,b}$ e cinco valores diferentes para $m_h = \{10^4, 10^5, 10^6, 10^7, 10^8\}$, em 50 amostras. É importante ressaltar que, obviamente, com o crescimento do valor sugerido para $m_h$ o número de amostras selecionadas, sem repetição, se aproxima muito do valor requerido de amostras ($n^{*,b}$).

Entretanto, existe a necessidade, no caso do exemplo em questão, do valor de $m_h$ atingir patamares acima de um bilhão de pacotes ($m_h = 10^8$) por estrato de tempo. Imaginando uma rede operando na faixa de 10 *Gbps*, para a qual se supõem um tamanho médio de pacotes igual a 1 *Kbyte*, seria necessário ter um intervalo para os estratos de aproximadamente 1000 segundos para, desta forma, obter-se um valor de $m_h$ acima de $10^8$. Embora este cenário seja plenamente factível em uma rede de alta capacidade e com alto tráfego agregado, no contexto de monitoramento e gerenciamento, a granularidade exigida para o tamanho dos estratos é relativamente alta, principalmente no provisionamento de QoS e controle de SLA´s.

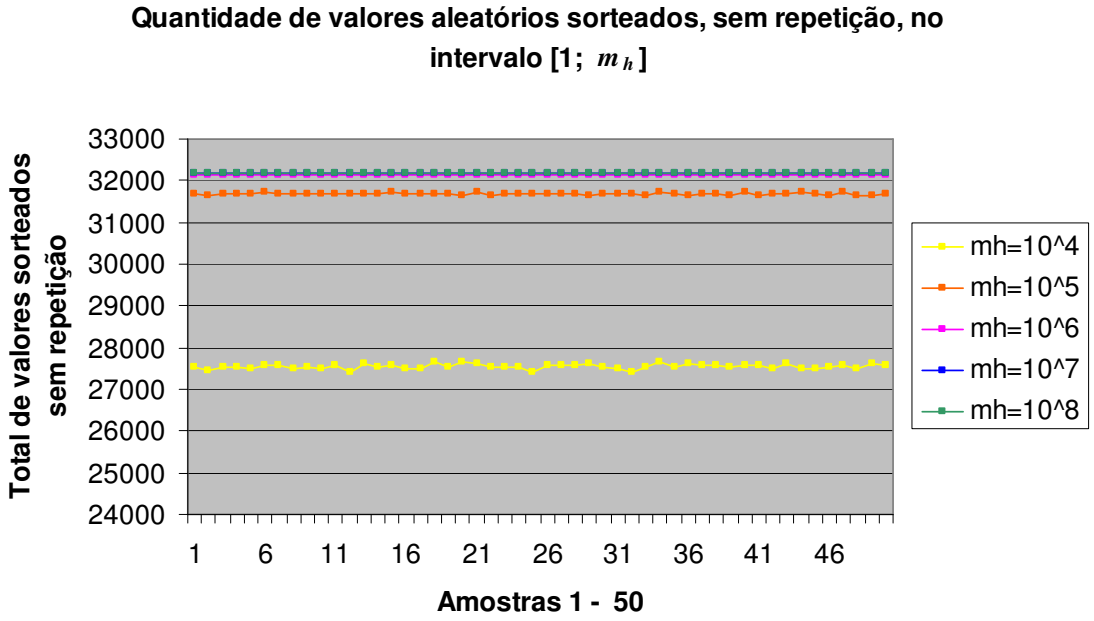



**Figura 22** Gráfico comparativo entre quantidade de números aleatórios sorteados, sem repetição, a partir dos valores sugeridos de $m_h$.

A Tabela 17 apresenta dados estatísticos com relação às informações apresentadas na Figura 22 , indicando, adicionalmente, o percentual de perda no número de amostras em relação ao número mínimo requerido ($n^{*,b}$), para os diferentes valores de $m_h$.

**Tabela 17** Estatísticas para os dados apresentados na Figura 22 .

| $m_h$ | Média aritmética | Desvio Padrão | Variância | Perda em relação a $n^{*,b}$ (%) |
|---|---|---|---|---|
| $10^4$ | 27544,8 | 59,21 | 3505,80 | 14,44 |
| $10^5$ | 31680,38 | 24,75 | 612,77 | 1,6 |
| $10^6$ | 32145,18 | 6,88 | 47,38 | 0,16 |
| $10^7$ | 32191,06 | 2,20 | 4,83 | 0,00015 |
| $10^8$ | 32195,42 | 0,61 | 0,37 | $\cong 0$ |

Para tratar esta restrição, optou-se por fazer um pequeno ajuste no algoritmo para geração dos números aleatórios. De forma geral, utilizou-se para o controle do número de iterações o comprimento da lista associativa, ou seja, constatando-se que a lista associativa cresce de acordo com a inserção dos valores sorteados, valores duplicados são descartados e sabendo que é necessário gerar um número $n^{*,b}$ de amostras aleatórias, basta manter um laço de repetição enquanto o número de elementos da lista for menor que o número mínimo de amostras requeridas ( $while(list.size() < n^{*,b})$ ).

Embora bastante simples de implementar, esta aproximação gerou um ponto de incerteza quanto ao seu desempenho. Para tal, efetuaram-se alguns testes utilizando os mesmos parâmetros do exemplo anterior, porém com foco no número de iterações necessárias para obter-se o número mínimo requerido de amostras aleatórias. Constatou-se que, para o valor de $m_h$ mais próximo $n^{*,b}$ é exigido o maior número de iterações. Entretanto, o número de iterações adicionais não representa aumento significativo no desempenho, principalmente, quando comparado à significativa perda no número de amostras, identificado no algoritmo anteriormente citado.

A Figura 23 apresenta o gráfico do número de iterações necessárias para atingir o número mínimo de amostras $n^{*,b}$, para os diferentes valores de $m_h$. A Tabela 18 apresenta dados estatísticos com relação às informações apresentadas na Figura 23 , indicando também, o percentual de aumento no número de iterações para atingir o número esperado de amostras.

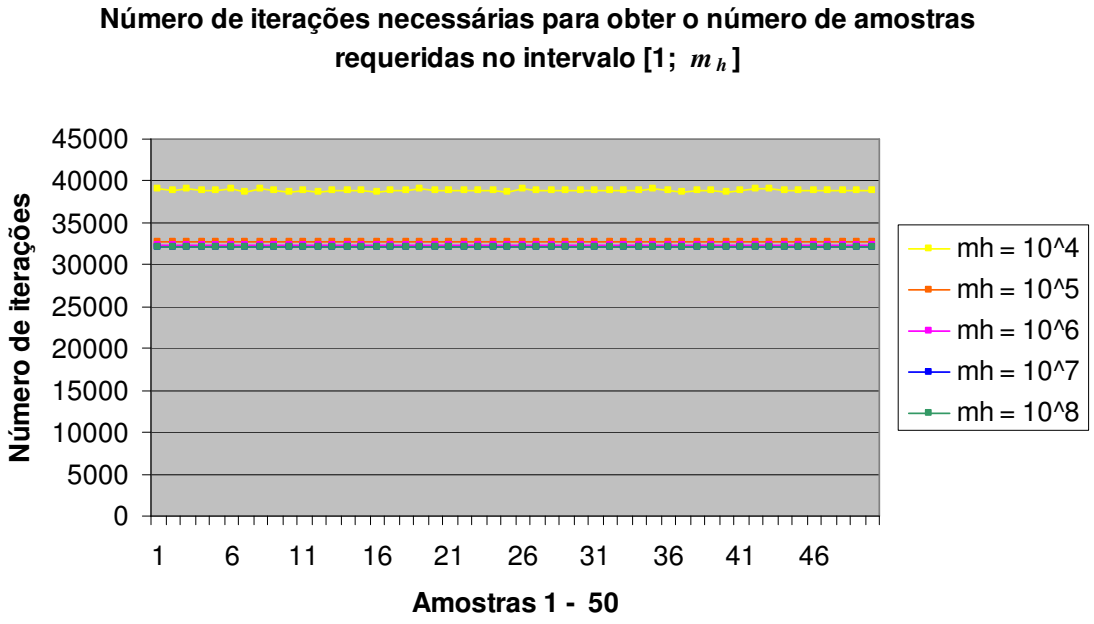



**Figura 23** Gráfico comparativo do número de iterações necessárias para atingir o número mínimo de amostras $n^{*,b}$, a partir dos valores sugeridos de $m_h$.

**Tabela 18** Estatísticas para os dados apresentados na Figura 23 .

| $m_h$ | Média | Desvio Padrão | Variância | Acréscimo no número de iterações (%) |
|---|---|---|---|---|
| $10^4$ | 38857,66 | 98,88 | 9777,74 | 17,1 |
| $10^5$ | 32727,1 | 23,91 | 571,68 | 1,6 |
| $10^6$ | 32249,76 | 6,94 | 48,19 | 0,16 |
| $10^7$ | 32201,14 | 2,31 | 5,35 | 0,00015 |
| $10^8$ | 32196,44 | 0,70 | 0,49 | $\cong 0$ |

A abordagem dessas questões relacionadas à geração dos valores aleatórios para composição das amostras foi resolvida de forma bastante simples, conforme citado anteriormente. Entretanto, é relevante ressaltar estes detalhes, pois na especificação da técnica de amostragem aleatória estratificada adaptativa os mesmos não são mencionados. Por este motivo, buscou-se enfatizar pontos de incerteza, possibilitando uma melhor compreensão da implementação.

Outro ponto que cabe ser destacado refere-se à implementação do mecanismo de tabelas temporárias de fluxo. A adoção deste mecanismo mostrou-se como uma forma simples e fácil de agregar as particularidades relacionadas à técnica de amostragem utilizada. Pelo fato de ter-se previamente implementada a estrutura de informações que compõem as estatísticas de fluxo, a qual foi consolidada em uma classe, bastou alocar um objeto temporário para cada estrato.

Além disso, esta forma de armazenar as informações em cada estrato possibilitou manter a estrutura de funcionamento do sistema de medição tradicional, requerendo apenas a adição do campo para contabilização do total de pacotes observados no decorrer da duração do fluxo. Ainda neste contexto, é interessante indicar o uso de uma *thread* auxiliar para o controle do período de duração de cada bloco.

Por fim, é conveniente casar a periodicidade de exportação para fluxos de grande duração com o intervalo indicado para a duração dos estratos, ou então utilizar um valor para o período de exportação maior que o período de duração de um estrato (neste caso, é recomendado valores múltiplos do período de duração de cada estrato). Isto pelo fato das estatísticas dos fluxos serem estimadas no final de cada bloco. Ou seja, se o período de exportação for menor que o período de duração de um estrato, serão exportadas informações redundantes, pois a tabela global de fluxos ainda terá as estatísticas estimadas no final do último bloco analisado.

## 5.2 AMOSTRAGEM ALEATÓRIA ESTRATIFICADA ADAPTATIVA

A presente seção tem por objetivo apresentar os resultados obtidos com relação aos procedimentos de teste realizados no decorrer do desenvolvimento do trabalho. Os procedimentos foram detalhados no Capítulo 4 , mais especificamente na seção 4.3. Inicialmente serão apresentados os resultados obtidos com relação à abordagem de predição, com o objetivo de identificar o grau de confiabilidade desta estimativa. Adicionalmente, serão apresentadas discussões com relação às estimativas do total de pacotes e do volume de *bytes* com o uso da técnica de amostragem.

### 5.2.1 Predição do total de pacotes para o próximo bloco

A predição do total de pacotes para o próximo bloco, utilizando o modelo Auto-Regressivo AR(1), foi alvo de testes no sentido de identificar o número de blocos passados necessários para diminuir o período de convergência em casos de oscilações significativas no tráfego. Para tal, foram utilizados os resultados dos testes efetuados nos procedimentos apresentados na Tabela 16 . A Figura 24 , a Figura 25 e a Figura 26 ilustram os resultados obtidos.

Embora a recomendação apresentada em [CHO 04][CHO 06] direcione para o uso do modelo AR(1) utilizando cinco valores passados, observou-se que em situações com alternância brusca nas condições do tráfego o modelo apresenta menor rigidez utilizando três blocos passados, gerando, conseqüentemente, uma convergência mais rápida. Esta constatação possibilita inferir que é desejável um conhecimento histórico mínimo do comportamento da rede em análise por parte do administrador, para que a configuração deste parâmetro seja realizada de forma condizente.

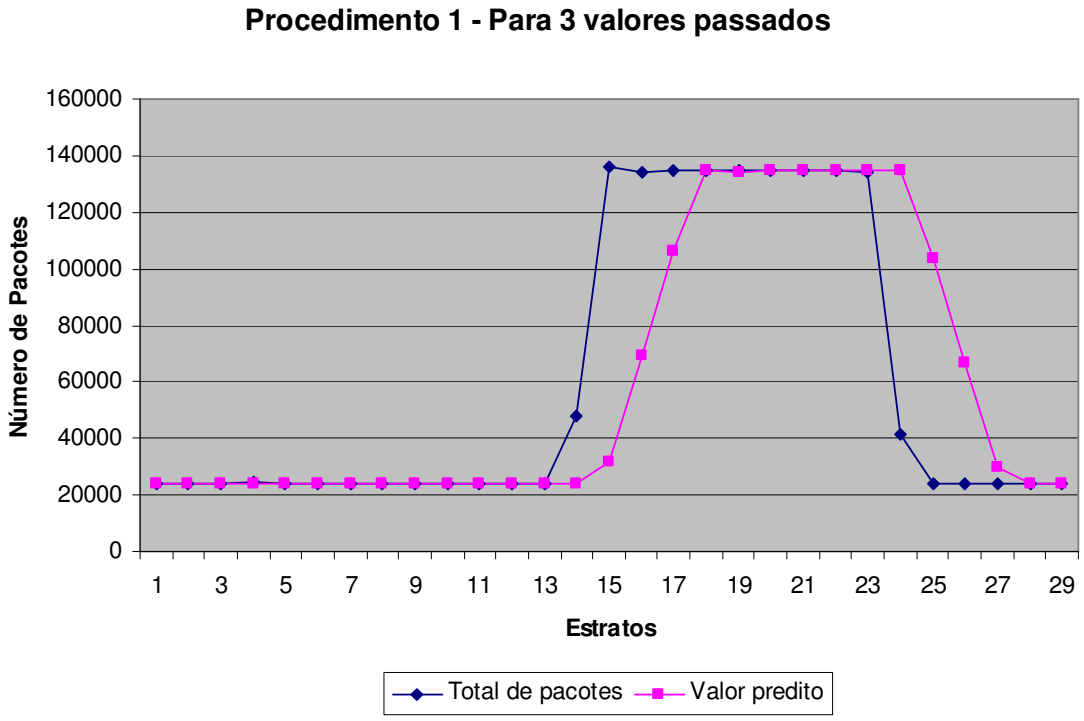



**Figura 24** Resultados obtidos para o procedimento 1.

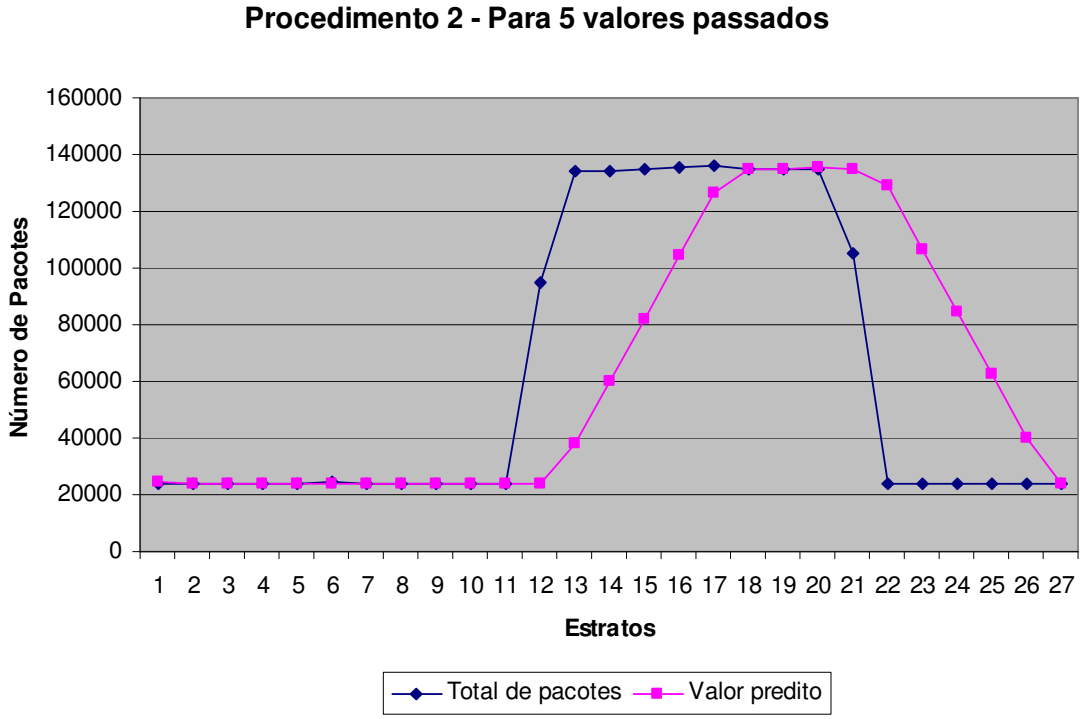



**Figura 25** Resultados obtidos para o procedimento 2.

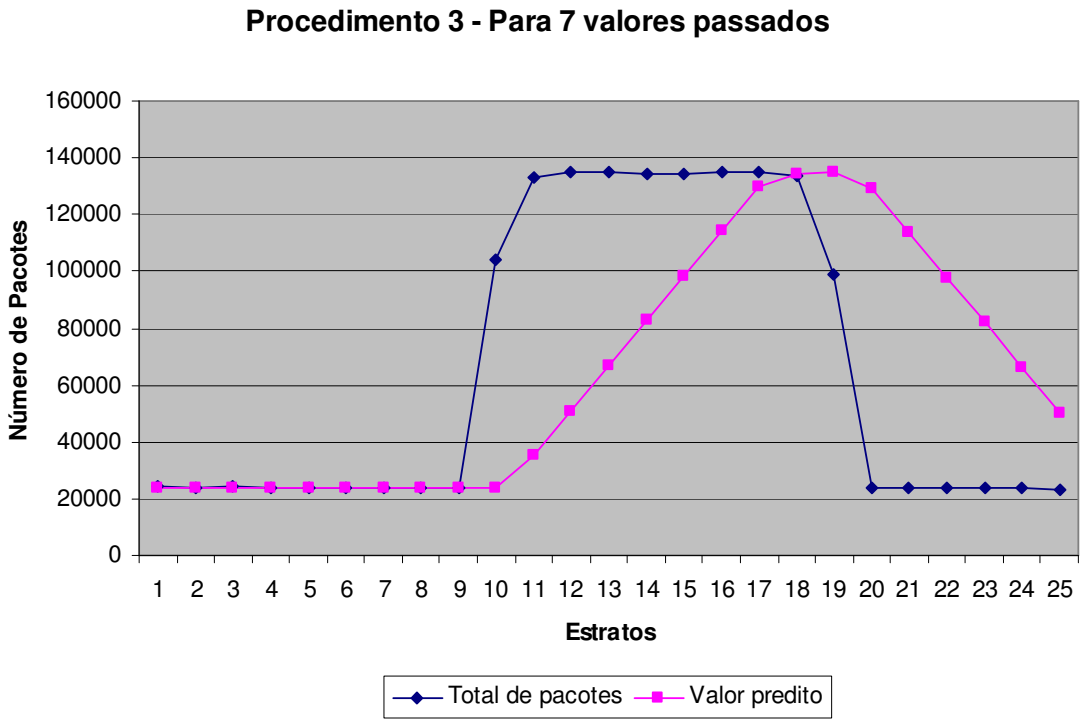



**Figura 26** Resultados obtidos para o procedimento 3.

5.2.2 Estimação do total de pacotes e do volume de *bytes*

Esta seção tem por objetivo apresentar as principais constatações a cerca dos resultados obtidos nos testes realizados. Seguindo a metodologia detalhada na seção 4.3.3, as discussões sobre os resultados serão realizadas com dois focos: o primeiro voltado aos resultados obtidos para o conjunto de teste no qual não se utilizou tráfego em *background*; o segundo voltado aos resultados oriundos do conjunto de teste no qual foi injetado tráfego competitivo. Os resultados obtidos nos procedimentos de teste do primeiro conjunto são apresentados através de gráficos e dados tabulares no APÊNDICE E.

Seguindo esta direção, a primeira consideração que pode ser efetuada com relação ao primeiro conjunto de teste, é a verificação da influência do tamanho dos estratos na fidedignidade das estimativas. Considerando que o primeiro conjunto de testes foi realizado utilizando blocos que compreendem intervalos de 10, 20 e 30 segundos, identificou-se que à

medida que o tamanho do bloco aumenta, os resultados das estimativas tendem a apresentar erros mais significativos.

Este fato se torna evidente no ambiente de teste utilizado, pois com o aumento no período de duração dos estratos, o tamanho da população aumenta proporcionalmente. Antes de prosseguir esta análise é conveniente analisar a Tabela 19 , a qual apresenta o tamanho médio da população (total de pacotes no intervalo de duração de um bloco) para os diferentes tamanhos de estrato utilizados.

**Tabela 19** Tamanho médio da população para os diferentes tamanhos de estrato.

| Duração dos estratos | Tamanho médio da população (número de pacotes) |
|---|---|
| 10 segundos | 133476,8182 |
| 20 segundos | 250733,17 |
| 30 segundos | 338842,5152 |

O comportamento no crescimento do tamanho da população é bastante particular do tipo de tráfego utilizado nos procedimentos de teste. Conforme explicitado na seção 4.3.2, utilizou-se para geração de fluxos de interesse, a transferência de arquivos sob o protocolo de aplicação HTTP. Este protocolo por sua vez utiliza para transporte dos dados o protocolo TCP, o qual possui nativamente em sua implementação a funcionalidade de controle de fluxo.

Esta característica do protocolo TCP, determinou que o tamanho da população fosse crescendo gradativamente com o aumento no tamanho dos estratos, uma vez que o controle de fluxo TCP ajusta o tamanho da janela de transmissão em razão das condições da rede. Desta forma, embora a transmissão dos fluxos de interesse tenha sido realizada de forma paralela e não-sincronizada, a capacidade de transmissão sempre esteve próxima do limiar máximo.

Nos instantes em que houve sobreposição nas transmissões dos fluxos de interesse, foi possível identificar claramente este comportamento, pois a taxa de transmissão era equalizada automaticamente pelo TCP para os diferentes fluxos, como forma de garantir o consumo ótimo do *link* de transmissão. Assim o volume de tráfego nos diferentes procedimentos de teste manteve-se muito semelhante nos estratos com mesma duração, apresentando um volume de pacotes também similar, fato que induz a inferência de que a variabilidade no tamanho dos pacotes foi pequena e que a taxa de amostragem era reduzida significativamente com o aumento no tamanho dos estratos. Este conjunto de fatores possibilitou entender a

diferença nos resultados das estimativas, quando do aumento no intervalo de duração dos estratos.

Outro ponto a ser destacado no primeiro conjunto de procedimentos de teste é relativamente mais visível e está relacionado aos valores atribuídos para os coeficientes $\eta$ e $\varepsilon$, confiabilidade e precisão das estimativas, respectivamente. A utilização de diferentes valores para estes coeficientes teve como objetivo verificar o comportamento da técnica de amostragem com a diminuição do número de amostras.

Neste contexto foi observado o comportamento esperado, ou seja, com a diminuição no número de amostras, concatenado ao aumento no tamanho dos estratos, o percentual de erro foi mais elevado. Os erros mais significativos nas estimativas aconteceram para os pequenos fluxos, chegando a patamares de mais de 100 % de erro, conforme o resultado obtido no procedimento 9. Para os grandes fluxos (fluxos "elefante"), o percentual de erro chegou a atingir 40 %, no procedimento 8.

Entretanto, é importante destacar a taxa de amostragem média (percentual de pacotes amostrados da população) para os diferentes tamanhos de estrato e número de amostras, nos diferentes procedimentos de teste. A Tabela 20 apresenta tais informações e a partir delas é possível verificar que mesmo na combinação do intervalo de menor duração (10 segundos) com o maior número de amostras ($\eta = 0.1$ e $\varepsilon = 0.1$), a taxa de amostragem pode ser considerada significativamente reduzida.

**Tabela 20** Taxa de amostragem média no primeiro conjunto de testes.

| | $n^{*,b}$ | Bloco (segundos) | Média da taxa de amostragem (%) |
|---|---|---|---|
| Procedimento 1 | 32196 | 10 | 24,1 % |
| Procedimento 2 | 32196 | 20 | 12,8 % |
| Procedimento 3 | 32196 | 30 | 9,5 % |
| Procedimento 4 | 4886 | 10 | 3,6 % |
| Procedimento 5 | 4886 | 20 | 1,9 % |
| Procedimento 6 | 4886 | 30 | 1,4 % |
| Procedimento 7 | 1420 | 10 | 1,1 % |
| Procedimento 8 | 1420 | 20 | 0,6 % |
| Procedimento 9 | 1420 | 30 | 0,4 % |

Conforme citado anteriormente, os resultados obtidos para o primeiro conjunto de procedimentos de teste, são apresentados no APÊNDICE E para facilitar a visualização devido à quantidade de informações. As medições extraídas do primeiro conjunto de teste foram essenciais para entender alguns aspectos intrínsecos a medição utilizando amostragem, as quais nortearam para os procedimentos do segundo conjunto de testes.

Neste sentido, utilizou-se o intervalo de duração para os estratos que se mostrou mais adequado (10 segundos) segundo o modelo de tráfego empregado, variando apenas os coeficientes $\eta$ e $\varepsilon$. Além disso, a presença de tráfego competitivo tinha por objetivo manter o ambiente de teste mais próximo de um ambiente real. A adição dos fluxos de 100 *Kbytes* e de 5 *Kbytes* foi efetuada com o intuito de manter um contraponto entre o número de fluxos "elefante" e de pequenos fluxos. Nos três procedimentos de teste a média do total de pacotes em cada bloco foi de 168130,6 pacotes. A taxa de amostragem média em cada um dos procedimentos de testes do segundo conjunto é apresentada na Tabela 21 .

**Tabela 21** Taxa de amostragem média no segundo conjunto de testes.

| | $n^{*,b}$ | Bloco (segundos) | Média da taxa de amostragem (%) |
|---|---|---|---|
| Procedimento 10 | 32196 | 10 | 19,2 % |
| Procedimento 11 | 4886 | 10 | 2,9 % |
| Procedimento 12 | 1420 | 10 | 0,8 % |

Com base nas informações apresentadas, os resultados obtidos com o uso da técnica de amostragem nos procedimentos do segundo conjunto de testes mostraram-se satisfatórios. A fidedignidade das estimativas para os fluxos "elefante", especificamente nos procedimentos

de teste 10 e 11, é plenamente aceitável, dado o percentual de pacotes amostrados. Especialmente no procedimento 11, onde a média da taxa de amostragem consolidou-se em torno de 2.9 % do total de pacotes em cada bloco, pode-se observar exatidão nas estimativas tanto do total de pacotes como do volume de *bytes*, sendo que, no pior caso, o percentual de erro não excedeu 2.28 %.

Para o procedimento 12, o percentual de erro começa apresentar valores mais significativos. Essa perda nas estimativas se apresenta de forma pontual, ou seja, não se distribui pelos fluxos, fato que induz a inferência de que a mínima variação na dinâmica de transmissão dos fluxos pode gerar distorções localizadas[6]. Além disso, pelos parâmetros configurados para este procedimento, no qual a taxa de amostragem foi reduzida significativamente, era esperado que os resultados das estimativas apresentassem divergências mais acentuadas.

De outro lado, as estimativas para os pequenos fluxos apresentaram-se bastante deturpadas nos três procedimentos, fato que demonstra a coerência no foco em grandes fluxos para o delineamento estatístico da técnica de amostragem empregada. A medição dos pequenos fluxos neste contexto, só foi realizada com o objetivo de ratificar este comportamento, pois em um ambiente real de medição os pequenos fluxos seriam descartados automaticamente, desalocando memória na tabela de fluxos e minimizando o transporte e armazenamento das informações de fluxo.

Abaixo são apresentados, em dados tabulares e gráficos, os resultados obtidos para cada um dos procedimentos de teste do segundo conjunto.

---

[6] Embora os testes tenham sido realizados em um ambiente controlado, o momento de transmissão dos fluxos de interesse pode ter sofrido variações mínimas nos diferentes procedimentos de teste. Esta característica pode ter contribuído para este comportamento de distorções pontuais nas estimativas.

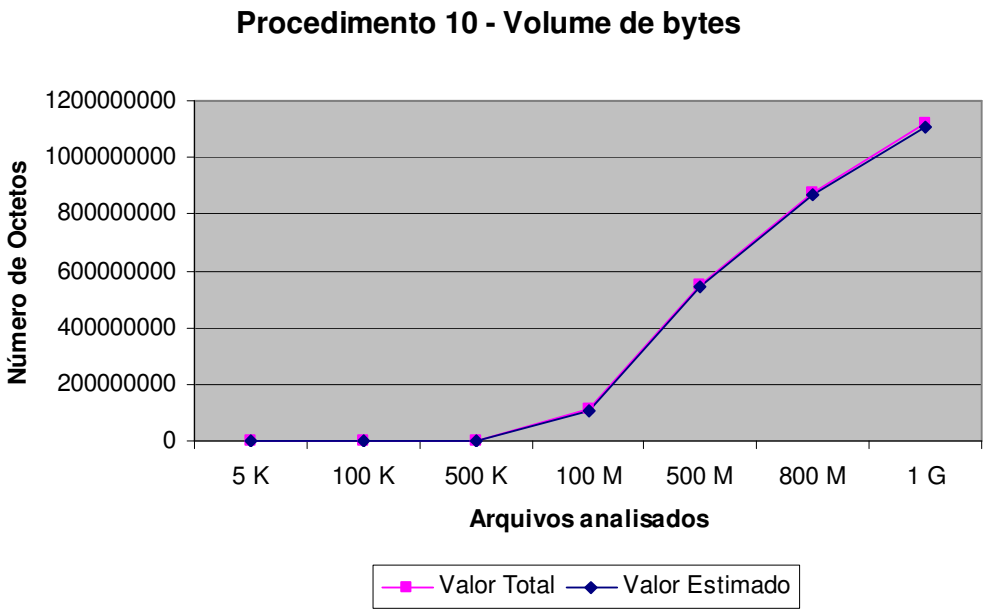



**Figura 27** Gráfico comparativo para o volume de *bytes* no procedimento 10.

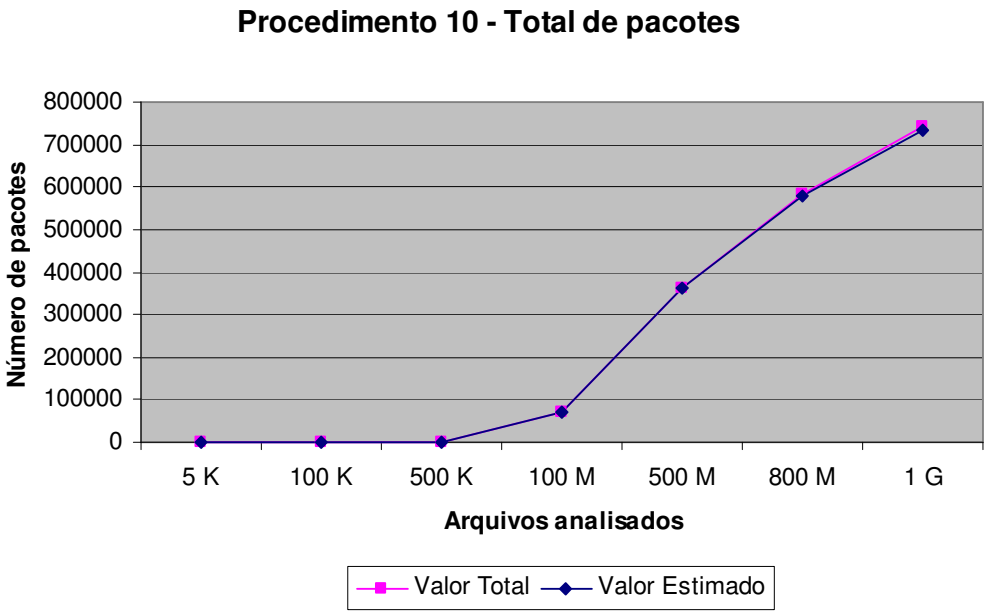



**Figura 28** Gráfico comparativo para o total de pacotes no procedimento 10.

**Tabela 22** Resultados para o volume de *bytes* no procedimento 10.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 5 K | 5893 | 3873 | 34,3 |
| 100 K | 107597 | 128750 | 19,7 |
| 500 K | 536008 | 0 | 100 |
| 100 M | 109660937 | 109144954 | 0,5 |
| 500 M | 548282818 | 544025363 | 0,8 |
| 800 M | 877253534 | 869895095 | 0,8 |
| 1 G | 1122882445 | 1105869892 | 1,5 |

**Tabela 23** Resultados para o total de pacotes no procedimento 10.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 5 K | 8 | 8 | 0 |
| 100 K | 75 | 89 | 18,7 |
| 500 K | 360 | 0 | 100 |
| 100 M | 72774 | 72420 | 0,5 |
| 500 M | 363554 | 360654 | 0,8 |
| 800 M | 581792 | 576920 | 0,8 |
| 1 G | 744551 | 733398 | 1,5 |

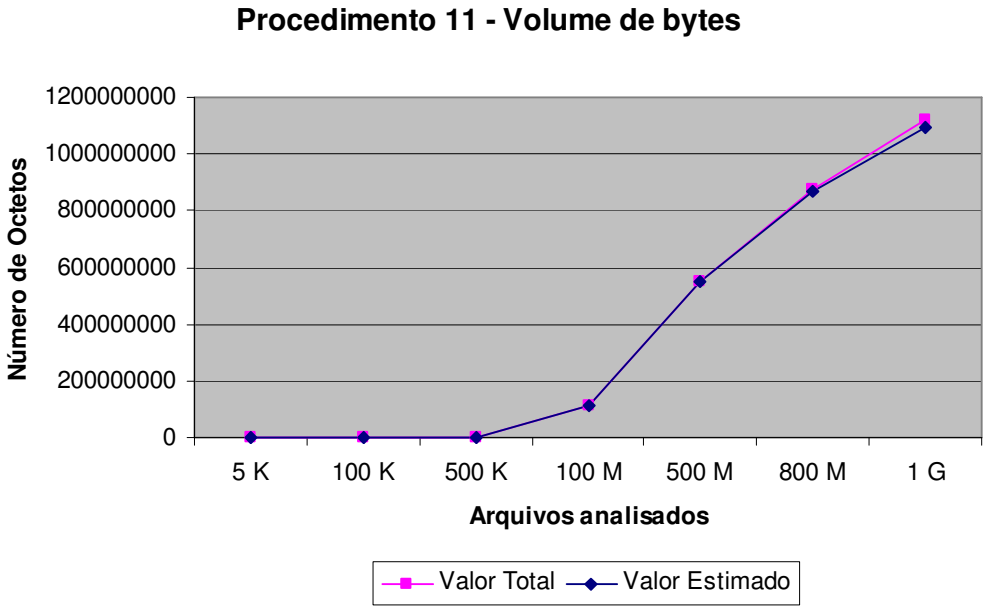



**Figura 29** Gráfico comparativo para o volume de *bytes* no procedimento 11.

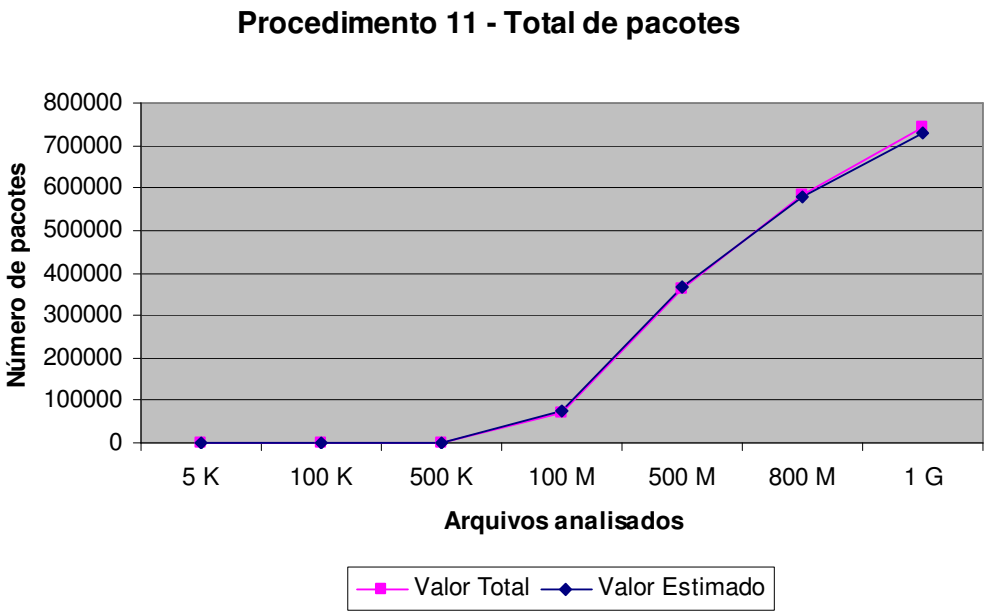



**Figura 30** Gráfico comparativo para o total de pacotes no procedimento 11.

**Tabela 24** Resultados para o volume de *bytes* no procedimento 11.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 5 K | 5893 | 25486 | 332,5 |
| 100 K | 107597 | 74063 | 31,2 |
| 500 K | 536074 | 561274 | 4,7 |
| 100 M | 109658693 | 112166769 | 2,3 |
| 500 M | 548292388 | 551017529 | 0,5 |
| 800 M | 877266058 | 871239122 | 0,7 |
| 1 G | 1122878315 | 1097074466 | 2,3 |

**Tabela 25** Resultados para o total de pacotes no procedimento 11.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 5 K | 8 | 30 | 275 |
| 100 K | 75 | 56 | 25,3 |
| 500 K | 361 | 397 | 10 |
| 100 M | 72740 | 74354 | 2,2 |
| 500 M | 363699 | 365689 | 0,5 |
| 800 M | 581894 | 577724 | 0,7 |
| 1 G | 744642 | 727738 | 2,3 |

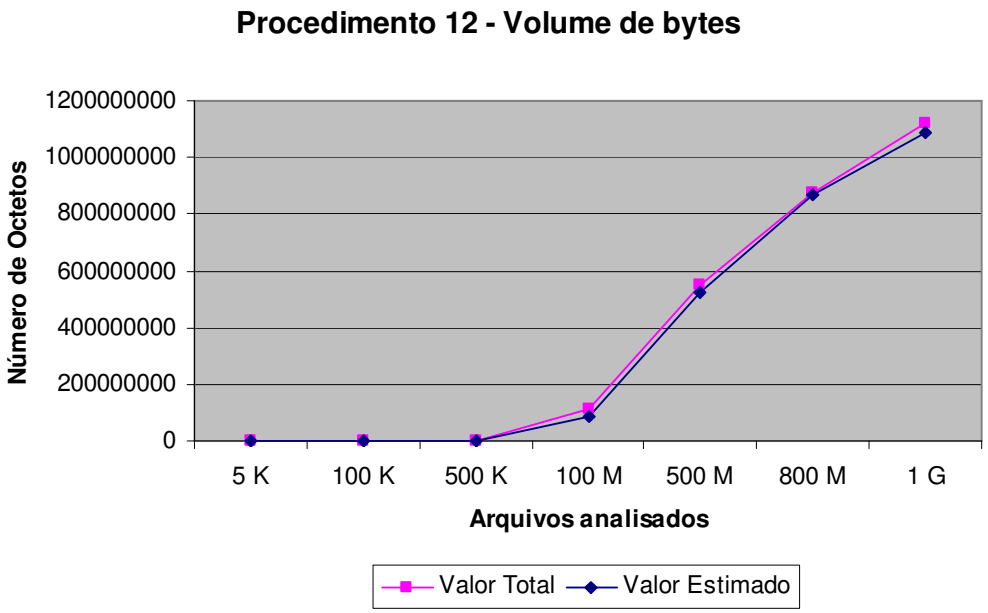



**Figura 31** Gráfico comparativo para o volume de *bytes* no procedimento 12.

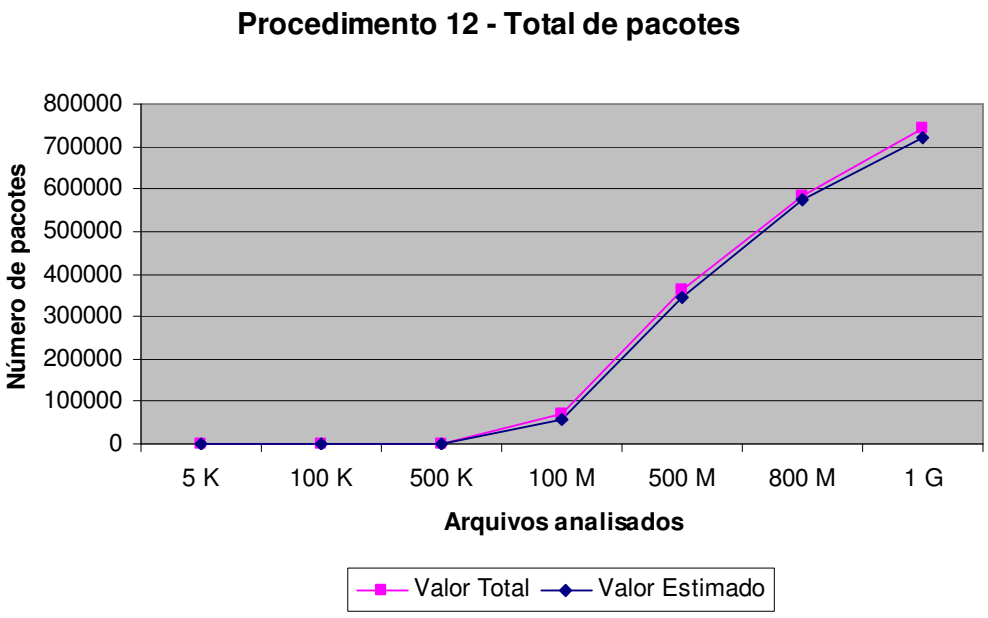



**Figura 32** Gráfico comparativo para o total de pacotes no procedimento 12.

**Tabela 26** Resultados para o volume de *bytes* no procedimento 12.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 5 K | 5893 | 83713 | 1320,5 |
| 100 K | 107597 | 255247 | 137,2 |
| 500 K | 536008 | 599448 | 11,8 |
| 100 M | 109661135 | 86313003 | 21,3 |
| 500 M | 548205558 | 520924836 | 5 |
| 800 M | 877179000 | 868843341 | 1 |
| 1 G | 1122829907 | 1088997520 | 3 |

**Tabela 27** Resultados para o total de pacotes no procedimento 12.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 5 K | 8 | 99 | 1137,5 |
| 100 K | 75 | 194 | 158,6 |
| 500 K | 360 | 489 | 35,8 |
| 100 M | 72777 | 57327 | 21,2 |
| 500 M | 363624 | 345624 | 5 |
| 800 M | 581825 | 576319 | 0,9 |
| 1 G | 744610 | 721734 | 3 |

# 6 CONCLUSÕES E TRABALHO FUTUROS

O presente capítulo tem por objetivo apresentar as conclusões relacionadas ao trabalho desenvolvido assim como algumas sugestões de trabalhos futuros.

## 6.1 CONCLUSÕES

A partir do desenvolvimento e dos resultados obtidos no presente trabalho, constatou-se que a técnica de amostragem aleatória estratificada adaptativa é aplicável à medição de tráfego baseada em fluxos e passível de ser implementada adicionalmente ao sistema tradicional de medição de tráfego. Entretanto, é importante destacar algumas constatações diagnosticadas no decorrer do desenvolvimento do trabalho.

Primeiramente, por tratar-se de um tema bastante inovador e que ainda germina no cenário científico, as proposições ainda estão no campo de pesquisa. Por este motivo, a busca por soluções nos pontos de incerteza foi bastante árdua, fato que demandou a inserção de mecanismos particulares para obtenção de um protótipo capaz de reproduzir as características especificadas pela técnica de amostragem. Neste contexto, é relevante destacar dois procedimentos em especial, os quais são explicitados a seguir.

O primeiro deles remete ao algoritmo utilizado para a geração dos valores aleatórios que, embora bastante simples, apresenta-se como uma solução eficiente para tratar a perda no número mínimo de amostras requeridas. Com a aplicação deste algoritmo, garantiu-se que o número mínimo de amostras requeridas pela técnica de amostragem estratificada adaptativa fosse extraído da população, possibilitando desta forma manter a implementação em conformidade com o delineamento estatístico da técnica de amostragem.

O mecanismo de tabelas temporárias utilizado para estimar as informações de fluxos de forma independente em cada estrato, também se apresenta como outro ponto a ser destacado. A aplicação deste mecanismo tornou factível a implementação, além de manter fielmente às

definições estabelecidas pela técnica de amostragem. Além disso, sua utilização apresentou um ganho significativo no processo de desenvolvimento propriamente dito, uma vez que a estrutura de dados utilizada para armazenamento das informações de fluxo pode ser reaproveitada, não impactando em alterações significativas no sistema tradicional de medição de tráfego baseada em fluxos.

Passando a avaliação da técnica de amostragem aleatória estratificada adaptativa, observou-se que a mesma deve ser empregada com base em um conhecimento prévio do comportamento usual da rede. Essa constatação foi extraída com base nos testes realizados, uma vez que variações significativas nos parâmetros de configuração, utilizando um mesmo ambiente de teste, apresentam, analogamente, variações bastante representativas nas estimativas obtidas.

Especificamente, observou-se que o intervalo de duração definido para o estrato pode gerar distorções nos resultados. Desta forma, o dimensionamento deste parâmetro deve ser realizado, primeiramente, com base na capacidade do enlace que está sendo mensurado. Adicionalmente, é importante levar em consideração, quando possível, as principais variáveis que possam gerar condições hostis flutuação e, consequentemente, comprometer a estimação das informações de fluxo. Este dimensionamento pode ser realizado de forma empírica, uma vez que o objetivo é minimizar possíveis perdas na fidedignidade das estimativas simplesmente por falta de critério e coerência na parametrização do sistema de medição.

Seguindo neste contexto, observou-se nos testes realizados que, quando parametrizado de forma condizente com o ambiente a ser mensurado, o sistema de medição utilizando a técnica de amostragem apresenta um erro percentual, para fluxos considerados “elefante”, que não ultrapassa 3% nas estimativas de contabilização do total de pacotes e volume de *bytes*. Este resultado mostra que a técnica de amostragem estratificada adaptativa, apresenta resultados plenamente aceitáveis para aplicação em diferentes contextos. Pelo erro percentual obtido, a utilização é factível até mesmo em cenários mais críticos como, por exemplo, a averiguação de quebra de cláusulas contratuais de um SLA.

Finalmente, outro ponto a ser destacado refere-se ao modelo temporal AR(1) utilizado para predição do total de pacotes. Basicamente, observou-se que o modelo AR(1) para cinco valores passados faz com que o ajuste da taxa de amostragem seja efetivamente adaptativo.

Entretanto, a partir dos testes realizados, foi possível observar que para condições de tráfego com oscilações drásticas, o modelo temporal AR(1) com três valores passados apresenta uma convergência maior que o modelo AR(1) para cinco e sete valores passados.

Este comportamento pode ser considerado característico, uma vez que o aumento no número de valores passados utilizado pelo modelo temporal ocasiona, diretamente, em uma maior rigidez no processo de convergência. Desta forma, este aspecto remete, novamente, a necessidade de conhecimento prévio do comportamento usual da rede a ser mensurada, fazendo com que este parâmetro seja configurado de forma a minimizar o tempo de convergência em redes que apresentam grandes flutuações, ou até mesmo maior estabilidade em ambientes com flutuações menores.

## 6.2 TRABALHOS FUTUROS

Como proposta de trabalhos futuros, projeta-se a implementação de outras técnicas de amostragem propostas no mesmo segmento do presente trabalho, no sentido de traçar uma análise comparativa.

Outra proposição que se apresenta com um campo promissor de pesquisa, diz respeito à implementação em *hardware* do sistema de medição como um todo, incluindo a abordagem tradicional e a aproximação utilizando amostragem. Além de seguir as recomendações do PSAMP, esta aproximação possibilitaria obter um ganho em desempenho bastante significativo, uma vez que seleção dos pacotes pode ser feita em nível de *hardware*, eliminando o processamento requerido dentre as etapas de captura e entrega dos pacotes a aplicação.

A elaboração de um projeto de experimentos também se apresenta como uma proposta bastante pertinente, uma vez que as iniciativas de implementar e avaliar diferentes modelos no contexto do presente trabalho, demanda, recorrentemente, a elaboração de ambientes de execução fundamentados e representativos.

# REFERÊNCIAS


[3GP 00] 3GPP. "**Study on PS domain services and capabilities**". 3GPP, TR-22.976, R2000-v2.0.0, dezembro de 2000.

[AQU 00] SALSANO, S.; RICCIATO, F.; WINTER, M.; EICHLER, G.; THOMAS, A.; FUENFSTUECK, F.; ZIEGLER, T.; BRANDAUER, C. "**Definition and usage of SLSs in the AQUILA consortium**". IETF *draft*, Novembro de 2000. Disponível por www em: <http://userver.ftw.at/~ziegler/draft-aquila-sls-00.txt>, último acesso 11/01/2007.

[AWD 02] AWDUCHE, D.; CHIU, A.; ELWALID, A.; WIDJAJA, I.; XIAO, X. "**Overview and Principles of Internet Traffic Engineering**". IETF - RFC 3272, Maio de 2002.

[AWD 99] AWDUCHE, D. "**MPLS and Traffic Engineering in IP Networks**", *IEEE Communications Magazine*, Vol. 37, Pág. 42 – 47, Dezembro de 1999.

[BRO 01] BROWNLEE, N. "**Using NeTraMet for Production Traffic Measurement**". *Integrated Network Management Proceedings*, *IEEE/IFIP International Symposium*. Maio de 2001.

[BRO 97a] BROWNLEE, N. "**Traffic Flow Measurement: Experiences with NeTraMet**". IETF - RFC 2123, Março de 1997.

[BRO 97b] BROWNLEE, N. "**Traffic Flow Measurement: Meter MIB**". IETF - RFC 2720, Outubro de 1999.

[BRO 99a] BROWNLEE, N.; MILLS, C.; RUTH G. "**Traffic Flow Measurement: Architecture**". IETF - RFC 2722, Outubro de 1999.

[CAL 05] CALYAM, P.; LEE, C.; ARAVA, P. K.; KRYMSKIY D.; LEE D. "**OnTimeMeasure: A Scalable Framework for Scheduling Active Measurements**". *End-to-End Monitoring Techniques and Services*, IEEE, Maio de 2005.

[CHO 04] CHOI, B.; PARK, J.; ZHANG, Z. "**Adaptive Packet Sampling for Accurate and Scalable Flow Measurement**". GLOBECOM'04, *IEEE Communications Society*, Dezembro de 2004.

[CHO 06] CHOI, B.; ZHANG, Z. "**Adaptive random sampling for traffic volume measurement**". *Telecommun Syst*, *Springer Science*. Dezembro de 2006.

[CIS 03] CISCO. "**Internetworking Technologies Handbook**".*Cisco System*, 2003. 1079p.

[CIS 06a] CISCO. "**Network Management System: Best Practises White Paper**". *Cisco System*, 2006. Disponível por wwww em: <http://www.cisco.com/warp/public/126/NMS_bestpractice.pdf>, último acesso em 11/01/2007.

[CIS 06b] CISCO. "**Cisco IOS NetFlow Overview**". *Cisco System*, Fevereiro de 2006. Disponível por wwww, último acesso em 11/01/2007.

[CLA 06] CLAISE, B. "**Packet Sampling (PSAMP) Protocol Specifications**". IETF *Draft*, Outubro de 2006. Disponível por www, último acesso 11/01/2007.

[CLA 93] CLAFFY, K. C.; POLYZOS, G. C.; BRAUN H. "**Application of Sampling Methodologies to Network Traffic Characterization**". *Proceedings ACM SIGCOMM'93*, *San Francisco*, *CA*, Setembro de 1993.

[COS 05] COSTA, R. I. T. "**Uma Arquitetura Para Apoio À Engenharia De Tráfego De Dados Nos Serviços De Comutação De Pacotes Dos Sistemas Celulares**".

Porto Alegre: PPGEE/PUCRS, 2005. 176p. (Dissertação de Mestrado)

[DER 00] DERI, L.; SUIN, E.; "**Effective Traffic Measurement Using nTop***". IEEE Communications Magazine*, Maio de 2000.

[DIE 06] DIETZ, T.; DRESSLER, F.; CARLE, G.; CLAISE, B.; AITKEN, P. "**Information Model for Packet Sampling Exports**". IETF *Draft*, Outubro de 2006. Disponível por www, último acesso 11/01/2007.

[DOV 04] DOVROLIS, C.; RAMANATHAN, P.; MORRE, D.; "**Packet Dispersion Techniques and Capacity Estimation**". *IEEE/ACM Transactions on Networking*, Vol. 12, Pág. 963 – 977, Dezembro de 2004.

[DOW 99] DOWNEY, A. B. "**Using pathchar to estimate Internet link characteristics**". *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, Pág. 241 - 250. 1999.

[DUF 00] DUFFIELD, N. G.; GROSSGLAUSER M. "**Trajectory sampling for direct traffic observation**". *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication ACM*, Pág. 271 – 282. 2000.

[DUF 05] DUFFIELD, N.; LUND, C.; THORUP M. "**Estimating Flow Distributions From Sampled Flow Statistics**". *IEEE/ACM Transactions on Networking*, Vol. 13, No. 5, Outubro de 2005.

[EST 03] ESTAN, C.; VARGHESE, G. "**New Directions in Traffic Measurement and Accounting: Focusing on the Elephants, Ignoring the Mice**". *ACM Transactions on Computer Systems*, Vol. 21, Pág. 270 – 313, Agosto de 2003.

[EST 04] ESTAN, C.; KEYS, K.; MOORE, D.; VARGHESE, G. "Building a Better NetFlow". *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications*, Pág. 245 – 256.

2004.

[FAN 99] FANG, W.; PETERSON, L. "**Inter-AS traffic patterns and their implications**". *Global Telecommunications Conference*, IEEE, Vol. 3, Pág. 1859 – 1868, Maio de 1999.

[FEL 01] FELDMANN, A.; GREENBERG, N.; REINGOLD, N.; LUND, C.; REXFORD, J.; TRUE, F. "**Deriving traffic demands for operational IP networks: Methodology and Experience**". *Transactions on Networking*, IEEE/ACM, Pág. 265 – 279, Junho de 2001.

[GAS 01] GASPARY, L.; BALBINOT, L. F.; STORCH, R.; WENDT, F.; TAROUCO, L. "**Uma Arquitetura para Gerenciamento Distribuído e Flexível de Protocolos de Alto Nível e Serviços de Rede**". Simpósio Brasileiro de Redes de Computadores - Florianópolis, SC, Brasil.2001.

[GEO 01] GEORGATOS, F.; GRUBER, F.; KARRENBERG, D.; SANTCROOS, M.; SUSANJ, A.; UIJTERWAAL, H.; WILHELM, R. "**Providing Active Measurements as a Regular Service for ISP´s**". *Proceedings of PAM*. 2001.

[HAN 99] HANDELMAN, S.; STIBLER, S.; BROWNLEE, N.; RUTH, G. "**RTFM: New Attributes for Traffic Flow Measurement**". IETF - RFC 2724, Outubro de 1999.

[HE 05] HE, G.; HOU J. C. "**An In-Depth, Analytical Study of Sampling Techniques for Self-Similar Internet Traffic**". icdcs, *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, Pág. 404-413. 2005.

[HEU 00] HEUSER, C. A. "**Projeto de Banco de Dados**". Porto Alegre: Sagra Luzzatto, 2000. 204p.

[HOH 06] HOHN, N.; VEITCH D. "**Inverting Sampled Traffic**". *IEEE/ACM*

*Transactions on Networking*, Vol. 14, No. 1, Fevereiro de 2006.

[ITU 02] ITU Y-1541. "**Series Y: Global Information Infrastructure And Internet Protocol Aspects**". *ITU-T Recommendation Y.1541*, Maio de 2002.

[IZK 06] IZKUE, E.; MAGAÑA, E. "**Sampling Time-Dependent Parameters in High-Speed Network Monitoring**". *Proceedings of the ACM international workshop on Performance monitoring, measurement, and evaluation of heterogeneous wireless and wired networks*, Pág. 13 - 17. 2006.

[JAC 89] JACOBSON, V.; LERES, C.; Mccanne, S. TCPDUMP. Disponível por www em: <http://www.tcpdump.org/> , último acesso em 12/01/2007.

[JED 92] JEDWAB, J.; PHALL, P.; PINNA B. "**Traffic estimation for the largest sources on a network using packet sampling with limited storage**". *Hewlett Packard*, 1992.

[JOR 00] JORMAKKA, J.; HEIKKINEN, K. "**QoS/GOS parameter definitions and measurements in IP/ATM networks**". *Springer Berlin / Heidelberg*, Vol. 1922/2000, Fevereiro de 2004.

[KAL 99] KALIDINDI, S.; ZEKAUSKAS, M. "**Surveyor: An infrastructure for internet performance measurements**". INET'99, *San Jose*, Junho de 1999.

[KAM 01] KAMIENSKI C. A.; SADOK D.; "**Chameleon: uma Arquitetura para Serviços Avançados Fim a Fim na Internet com QoS**". Simpósio Brasileiro de Redes de Computadores - Florianópolis, SC, Brasil.2001.

[KAM 05] KAMIENSKI, C.; SOUZA, T.; FERNANDES, S.; SILVESTRE, G.; SADOK, D. "**Caracterizando Propriedades Essenciais do Tráfego de Redes através de Técnicas de Amostragem Estratificada**". Simpósio Brasileiro de Redes de Computadores – Fortaleza, CE, Brasil. Maio de 2005.

[LI 04] LI, J.; SUNG, M.; XU, J.; LI, L. "**Large-Scale IP Traceback in High-Speed Internet Pratical Techniques and Theoretical Foundation**". *Proceedings of the IEEE Symposium on Security and Privacy*.2004.

[MAN 04] MANDRAWA, W.; CALYAM, P.; SRIDHARAN, M.; KHAN, A.; SCHOPIS, P.; "**H.323 Beacon: An H.323 application related end-to-end performance troubleshooting tool**". *Proceedings of the ACM SIGCOMM workshop on Network troubleshooting: research, theory and operations practice meet malfunctioning reality*, Pág. 241 – 246. 2004.

[MAO 03] MAO, Z.; BUSH, R.; GRIFFIN, T.; ROUGHAN, M. "**BGP Beacons**". *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, Pág. 1 – 14. 2003.

[MAR 02] MARILLY E.; MARTINOT O.; BETGÉ-BREZETZ S.; DELÈGUE G. "**Requirements for Service Level Agreement Management**". *IP Operations and Management,* IEEE, Pág 57- 62, Dezembro de 2002.

[MAR 06] MARK, L. "**libIPFIX: Internet Measurement Project**". Disponível por www em: <http://ants.fokus.fraunhofer.de/libipfix/>, último acesso em 12/01/2007.

[MAT 98] MATSUMOTO, M.; NISHIMURA, T. "**Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator**". *ACM Transactions on Modeling and Computer Simulation*, Vol. 8, No. 1, Pág. 3–30, Janeiro de 1998.

[MCC 93] McCANNE, S.; JACOBSON, V. "**The BSD Packet Filter: A New Architecture for User-level Packet Capture**". *Winter USENIX conference*, *San Diego*, CA, Janeiro de 1993.

[MCG 00] MCGREGOR T.; BRAUN, H.-W.; BROWN, J. "**The nlar network analysis infrastructure**". *IEEE Communication Magazine*, vol. 38, no. 5, Maio de 2000.

[MOR 01] MORRIS, R.; STURM, W.; JANDER, M. "**Service Level Managemen**t". Fundamentos do Gerenciamento de Níveis de Serviço. Rio de Janeiro. Editora Campus. 2001.

[MOR 04] MORI, T.; UCHIDA, M.; KAWAHARA, R. "**Identifying Elephant Flows Through Periodically Sampled Packets**". *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, Pág. 115 – 120. 2004.

[NIC 04] NICCOLINI, S.; MOLINA, M.; RASPALL, F.; TARTARELLI, S. "**Design and implementation of a One Way Delay passive measurement system**". *Network Operations and Management Symposium*, IEEE/IFIP, Vol. 1, Pág. 469- 482, Agosto de 2004.

[PAX 98] PAXSON, V.; ALMES, G. ; MAHDAVI, J. ; MATHIS, M. "**Framework for IP Performance Metrics**". IETF - RFC 2330, Maio de 1998.

[PHA 01] PHAAL, P.; PANCHEN, S.; McKEE, N. "**InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks**". IETF - RFC 3176, Setembro de 2001.

[PLO 06] PLONKA, D.; BROWNLEE, N. "**IP Flow Information Export (ipfix)**". IETF *charter*. Disponível por www em: <http://www.ietf.org/html.charters/ipfix-charter.html>, último acesso em 12/01/2007.

[QUI 06] QUITTEK, J. "**Packet Sampling** (psamp)". IETF *charter*. Disponível por www em: <http://www.ietf.org/html.charters/psamp-charter.html> último acesso em 12/01/2007.

[RÄI 03] RÄISÄNEN, V. "**Implementing Service Quality in IP NetWorks**". Editora *Wiley*, 2003. 325p.

[SAN 07a] SANTOS, G. L. "**Sistema Baseado nas Recomendações IPFIX para exportação e análise de informações de fluxos em redes convergentes**". Porto Alegre: PPGEE/PUCRS, 2004. 185p. (Dissertação de Mestrado)

[SAN 07b] SANTOS, G.; GUIMARÃES, V.; SILVEIRA, J.; VIEIRA, A.; NETO, J. A.; COSTA, R.; BALBINOT, R. "**UAMA: a Unified Architecture for Active Measurements in IP Networks**". *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*. Maio de 2007.

[SCH 98] SCHILDT, H. "**C++: The Complete Reference Third Edition**". *Osborne McGraw-Hill - New York*, Pág. 255.1998.1041p.

[SFL 03] SFLOW. "**Traffic Monitoring using sFlow**". Disponível por www em: <http://www.sflow.org>, último acesso em 12/01/2006.

[SHA 04] SHALUNOV, S.; TEITTELBAUM, B. "**One-way Active Measurement Protocol (OWAMP)**". IETF - RFC 3763, Abril de 2004.

[THO 97] THOMPSON, K.; MILLER, G. J.; WILDER, R. "**Wide-area Internet traffic patterns and characteristics**". IEEE *Network*, Vol. 11, Pág. 10 – 23, Dezembro de 1997.

[TRI 02] TRIMINTZIOS P.; ANDRIKOPOULOS I.; PAVLOU G.; FLEGKAS P.; GRIFFIN D.; GEORGATSOS P.; GODERIS D.; GEORGIADIS L.; JACQUENET C.; EGAN R. "**A Management and Control Architecture for Providing IP Differentiated Services in MPLS-based Networks**". *IEEE Communication Magazine*, Vo.l. 38, Pág. 80-88, Agosto de 2002.

[VIE 04] VIEIRA, A. T. "**Pesquisa, Desenvolvimento e Construção de uma Ferramenta para Gerência de Desempenho em Redes Convergentes Baseada na Medida em Tempo Real do Tráfego Classificado por Fluxos**". Porto Alegre: PPGEE/PUCRS, 2004. 185p. (Dissertação de Mestrado)

[WAL 00] WALDBUSSER, S. "**Remote Network Monitoring Management Information Base**". IETF - RFC 2819, Maio de 2000.

[WAN 01] WANG, Z. "**Internet QoS – Architetures and Mechanisms for Quality of Service**". *Morgan Kaufmann*. 2001. 239p.

[XU 05] XU, L.; WU, G.; LI J. "**Packet-Level Adaptive Sampling on Multi-Fluctuation Scale Traffic**". *Proceedings Communications, Circuits and Systems*, Vol. 1, Pág. 604- 608, Agosto de 2005.

[ZHA 02] ZHANG, Y.; BRESLAU, L.; PAXSON, V.; SHENKER, S. "**On the Characteristics and Origins of Internet Flow Rates**". *Proceedings of ACM SIGCOMM*, Pág. 309 –322, Agosto de 2002.

[ZHA 04] ZHANG, F.; LEI, Z. "**The Evaluation of Poisson Packet Sampling Measurement Techniques**". *5th International Symposium on Multi-Dimensional Mobile Communications Proceedings*, IEEE CNF, Vol. 1, Pág. 239 – 243, Setembro de 2004.

[ZSE 05] ZSEBY, T.; MOLINA, M.; DUFFIELD, N.; NICCOLINI, S.; RASPALL, F. "**Sampling and Filtering Techniques for IP Packet Selection**". IETF *Draft*, Julho de 2005. Disponível por www, último acesso 12/01/2007.

# APÊNDICE A – Algoritmos para geração de números aleatórios na técnica de amostragem avaliada.

Abaixo é apresentada a troca de *e-mail* realizada com a autora da técnica de amostragem avaliada no presente trabalho. A mensagem foi escrita com o objetivo de averiguar se, nos experimentos realizados pelos autores, utilizou-se algum algoritmo específico para geração de números aleatórios.

Pergunta:

*... In the figure 2 of the paper, you show the flow chart of the adaptive random sampling procedure. In the stage "Randomly sample incoming packets with Psp", do you use a specific random function (for example: Mersenne twister, RANROT-B, RANROT-W, Mother-of-all ...) ? ...*

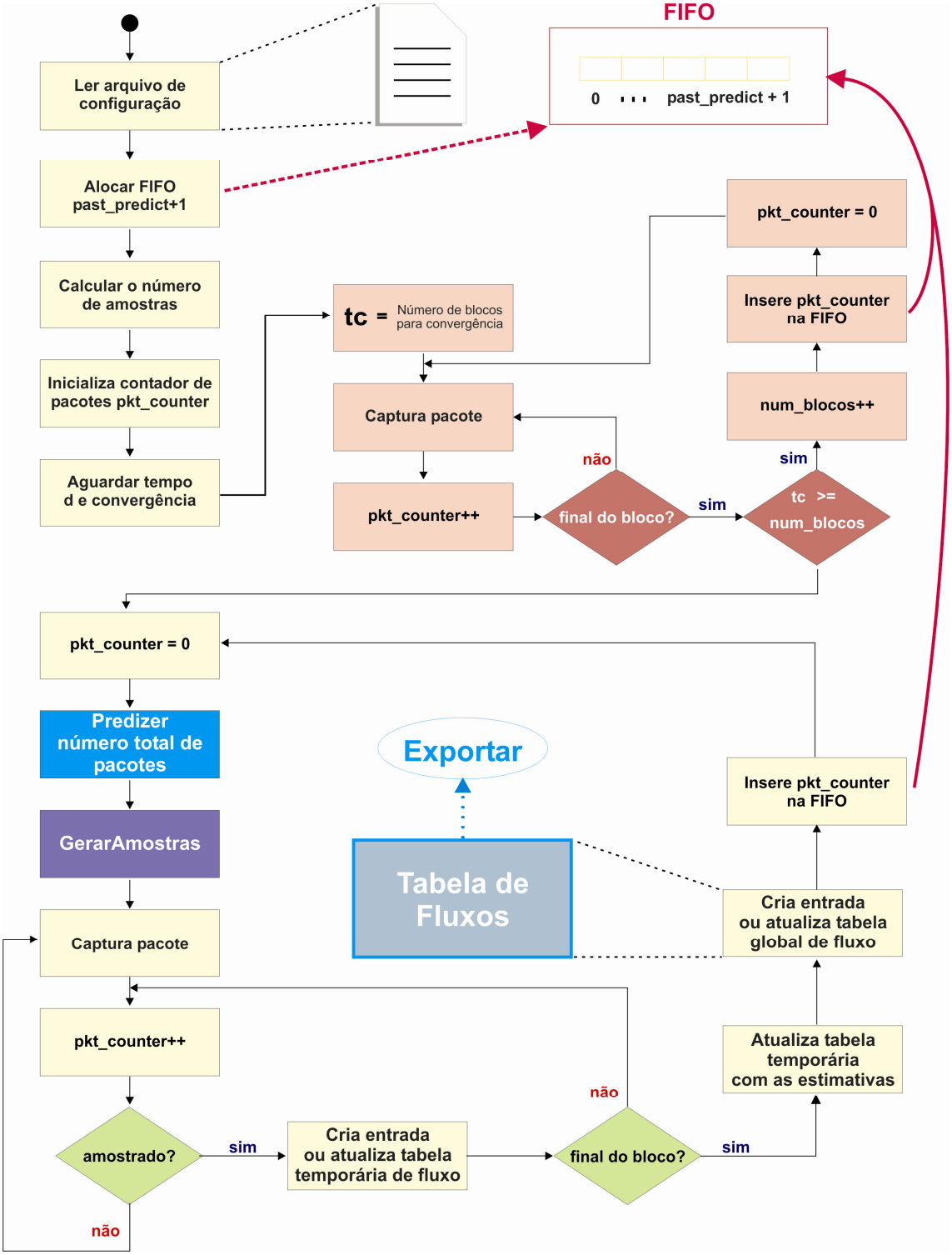
Resposta:

*You can use a random function provided by any programming language.*

*Thanks,*

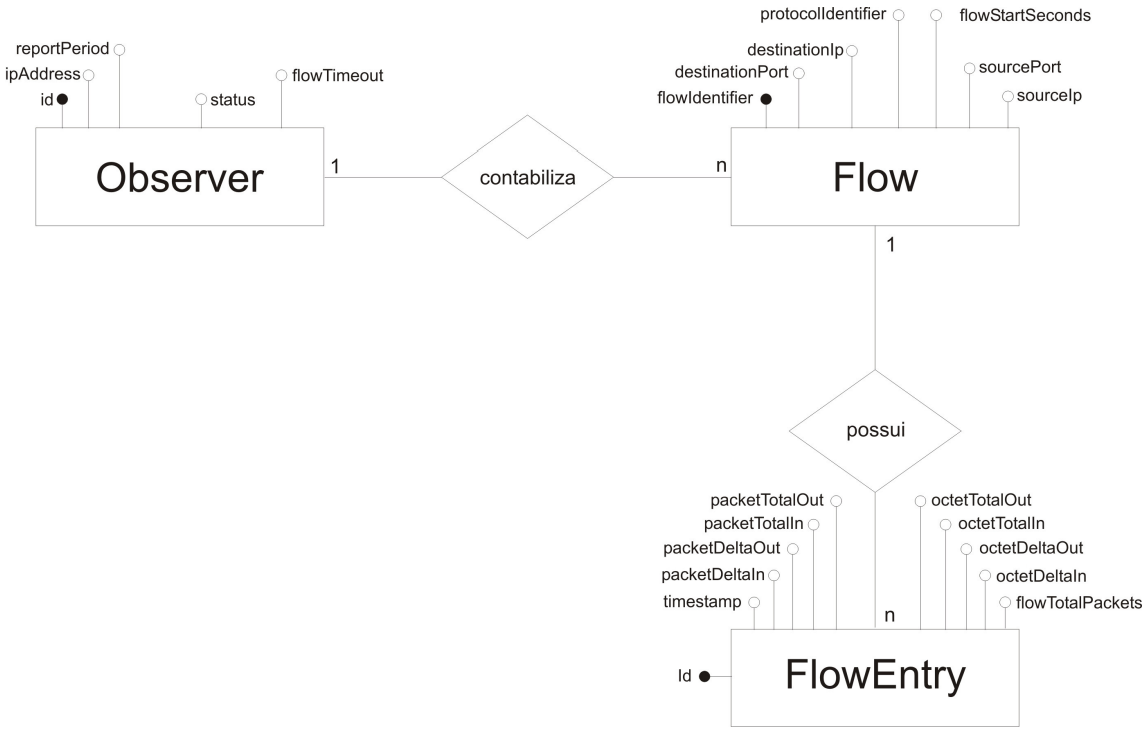
*--Baek-Young*

APÊNDICE B – Fluxo geral de funcionamento do sistema de medição utilizando amostragem



APÊNDICE C – Modelo ER da base de dados

Abaixo é apresentado o modelo ER da base de dados, a qual foi utilizada para o armazenamento das informações de fluxo.



# APÊNDICE D – Código-fonte do modelo AR(1) e seleção de amostras

```
size_t PredictSampler::IndexSamples(std::set<size_t> &sample_space) {
std::list<size_t> win_hist_temp;
win_hist_temp = win_hist_size;
win_hist_temp.pop_back();
double average_size, average_temp;
average_size = std::accumulate(++win_hist_size.begin(), win_hist_size.end(), 0) / (win_hist_size.size() - 1);
average_temp = std::accumulate(win_hist_temp.begin(), win_hist_temp.end(), 0) / win_hist_temp.size();
double sqxy = 0;
std::list<size_t>::iterator it_size = ++win_hist_size.begin();
std::list<size_t>::iterator it_temp = win_hist_temp.begin();
for (int i = 0; i < (win_hist_size.size() - 1); ++i) {
        sqxy += ((*it_temp) - average_temp) * ((*it_size) - average_size);
        ++it_size;
        ++it_temp;
        }
double sqx = 0;
for (it_temp = win_hist_temp.begin(); it_temp != win_hist_temp.end(); ++it_temp)
        sqx += pow((*it_temp) - average_temp, 2);
double a, b;
int64_t mh;
b = (sqxy / ((sqx == 0) ? 1 : sqx));
a = average_size - (b * average_temp);
mh = (uint64_t)(a + (b * win_hist_size.back()));
struct timeval tv;
gettimeofday(&tv, NULL);
TRandomMersenne rg(tv.tv_usec);
sample_space.clear();
if (mh > sampling_count) {
        for (int i = 0; i < sampling_count; ++i)
                sample_space.insert(rg.IRandom(1, mh));
        full_sampling = false;
        }
else
        full_sampling = true;
return sample_space.size();
};
```

APÊNDICE E – Resultados do primeiro conjunto de procedimentos de teste

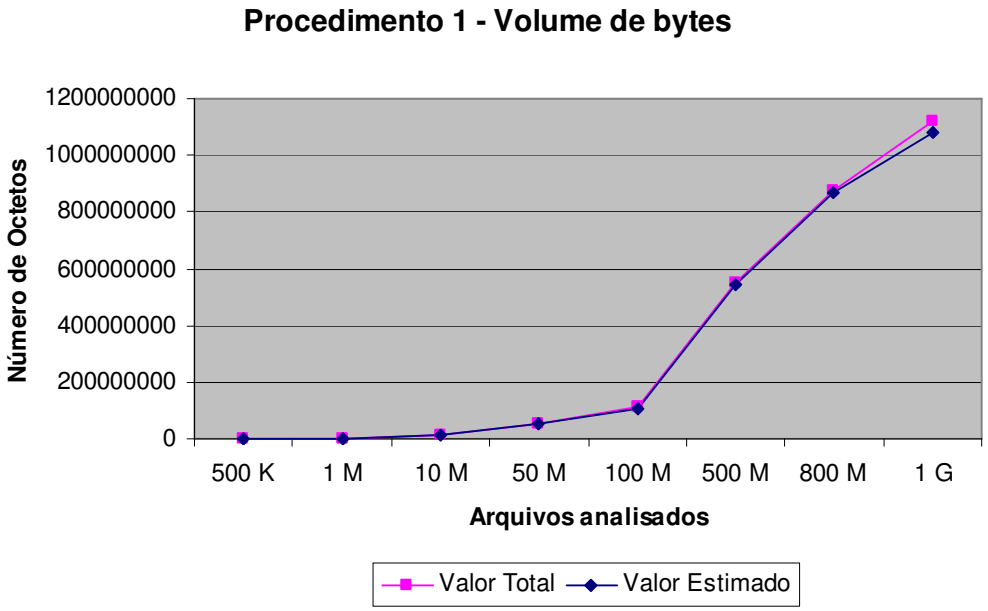



**Figura 33** Gráfico comparativo para o volume de *bytes* no procedimento 1.

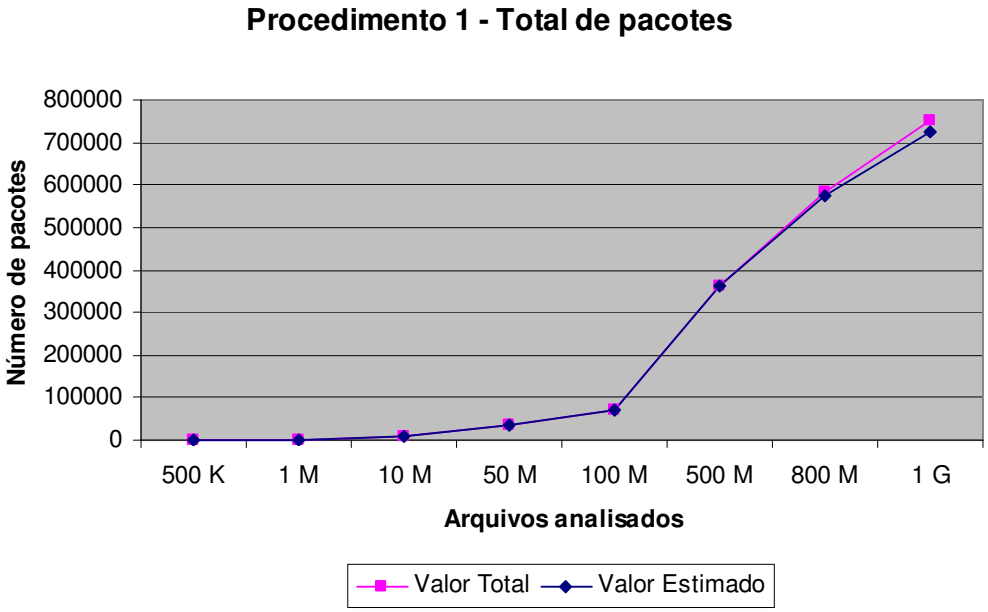



**Figura 34** Gráfico comparativo para o total de pacotes no procedimento 1.

**Tabela 28** Resultados para o volume de *bytes* no procedimento 1.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 536074 | 533470 | 0,5 |
| 1 M | 1097072 | 1220856 | 11,3 |
| 10 M | 10951355 | 10822172 | 1,2 |
| 50 M | 54830000 | 55132404 | 0,6 |
| 100 M | 109590669 | 108732920 | 0,8 |
| 500 M | 547859878 | 542915605 | 0,9 |
| 800 M | 876839300 | 866144584 | 1,2 |
| 1 G | 1123287751 | 1080528330 | 3,8 |

**Tabela 29** Resultados para o total de pacotes no procedimento 1.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 361 | 362 | 0,3 |
| 1 M | 731 | 810 | 10,8 |
| 10 M | 7292 | 7200 | 1,3 |
| 50 M | 36378 | 36567 | 0,5 |
| 100 M | 72916 | 72329 | 0,8 |
| 500 M | 363852 | 360537 | 0,9 |
| 800 M | 582247 | 575075 | 1,2 |
| 1 G | 751816 | 723380 | 3,8 |

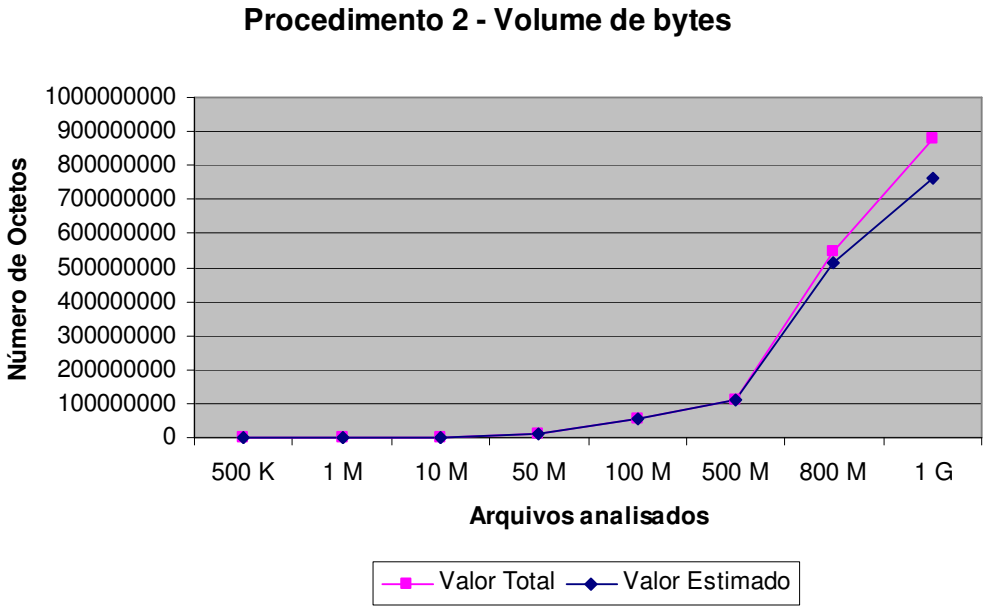



**Figura 35** Gráfico comparativo para o volume de *bytes* no procedimento 2.

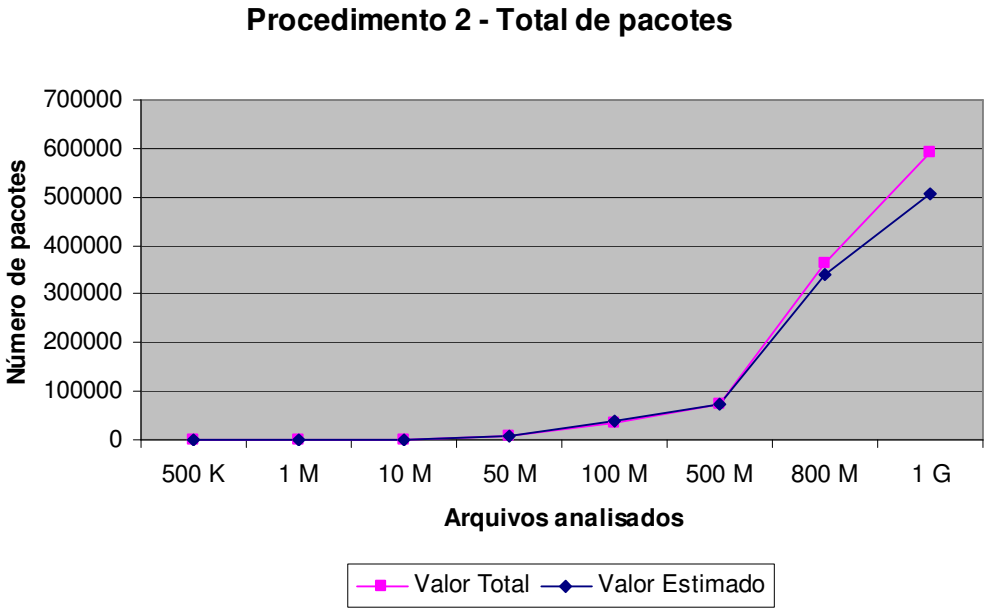



**Figura 36** Gráfico comparativo para o total de pacotes no procedimento 2.

**Tabela 30** Resultados para o volume de *bytes* no procedimento 2.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 536074 | 478882 | 10,7 |
| 1 M | 1097204 | 1174775 | 7,1 |
| 10 M | 10876385 | 10920260 | 0,4 |
| 50 M | 54683548 | 55629393 | 1,7 |
| 100 M | 109307811 | 109182447 | 0,1 |
| 500 M | 546907886 | 513023388 | 6,2 |
| 800 M | 877841730 | 763236838 | 13,1 |
| 1 G | 1073460315 | 1035130197 | 3,6 |

**Tabela 31** Resultados para o total de pacotes no procedimento 2.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 361 | 337 | 6,6 |
| 1 M | 733 | 800 | 9,1 |
| 10 M | 7255 | 7277 | 0,3 |
| 50 M | 36380 | 37028 | 1,8 |
| 100 M | 72683 | 72622 | 0,1 |
| 500 M | 363952 | 341716 | 6,1 |
| 800 M | 591818 | 507721 | 14,2 |
| 1 G | 714362 | 688917 | 3,6 |

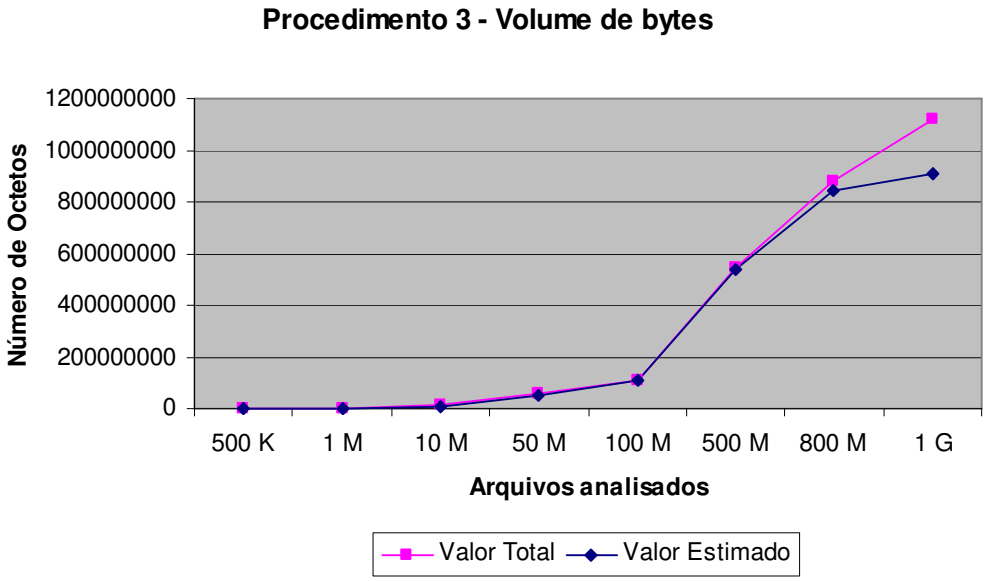



**Figura 37** Gráfico comparativo para o volume de *bytes* no procedimento 3.

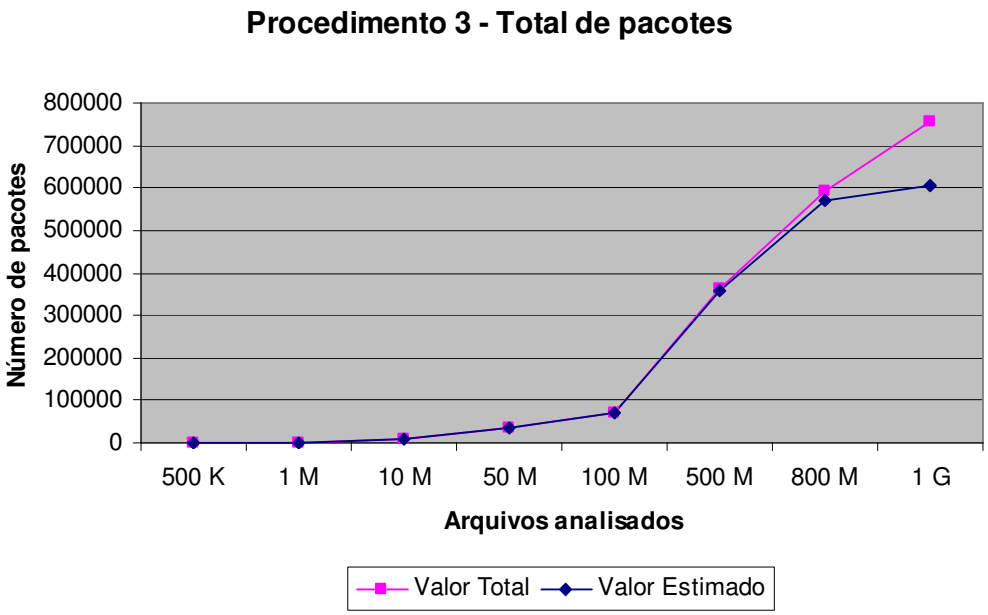



**Figura 38** Gráfico comparativo para o total de pacotes no procedimento 3.

**Tabela 32** Resultados para o volume de *bytes* no procedimento 3.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 536206 | 507078 | 5,4 |
| 1 M | 1097534 | 1164267 | 6,1 |
| 10 M | 10969395 | 10021134 | 8,6 |
| 50 M | 54828350 | 54189904 | 1,2 |
| 100 M | 109568929 | 105490446 | 3,7 |
| 500 M | 547148504 | 538350445 | 1,6 |
| 800 M | 877410120 | 843553539 | 3,9 |
| 1 G | 1123130553 | 905785634 | 19,4 |

**Tabela 33** Resultados para o total de pacotes no procedimento 3.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 363 | 351 | 3,3 |
| 1 M | 738 | 789 | 6,9 |
| 10 M | 7324 | 6677 | 8,8 |
| 50 M | 36353 | 35889 | 1,3 |
| 100 M | 72806 | 70076 | 3,7 |
| 500 M | 363897 | 358123 | 1,6 |
| 800 M | 591193 | 568067 | 3,9 |
| 1 G | 755701 | 603730 | 20,1 |

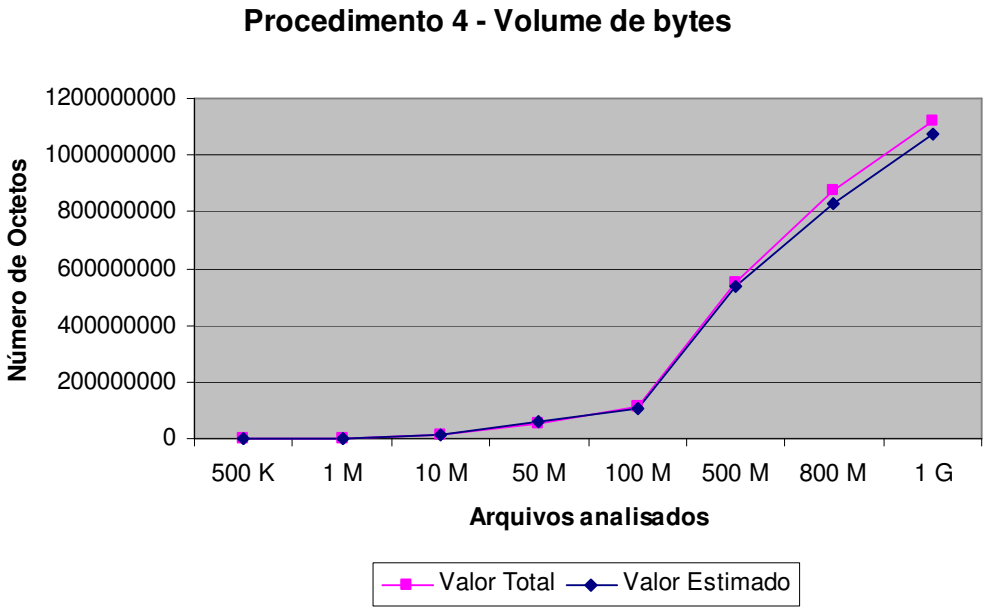



**Figura 39** Gráfico comparativo para o volume de *bytes* no procedimento 4.

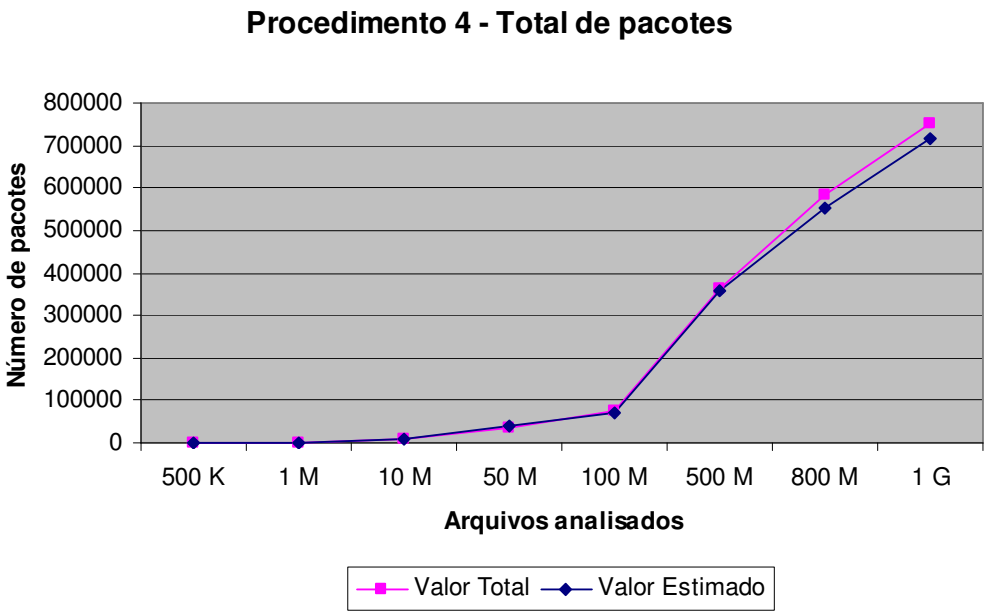



**Figura 40** Gráfico comparativo para o total de pacotes no procedimento 4.

**Tabela 34** Resultados para o volume de *bytes* no procedimento 4.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 536008 | 376975 | 29,7 |
| 1 M | 1097204 | 1129825 | 3 |
| 10 M | 10965501 | 10953198 | 0,1 |
| 50 M | 54831716 | 56931890 | 3,8 |
| 100 M | 109564861 | 107901251 | 1,5 |
| 500 M | 547522134 | 539201653 | 1,5 |
| 800 M | 875067852 | 831526686 | 5 |
| 1 G | 1121442805 | 1074860805 | 4,2 |

**Tabela 35** Resultados para o total de pacotes no procedimento 4.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 360 | 275 | 23,6 |
| 1 M | 733 | 772 | 5,3 |
| 10 M | 7265 | 7324 | 0,8 |
| 50 M | 36404 | 37794 | 3,8 |
| 100 M | 73108 | 71952 | 1,6 |
| 500 M | 364020 | 358397 | 1,5 |
| 800 M | 581731 | 552933 | 5 |
| 1 G | 749267 | 718217 | 4,1 |

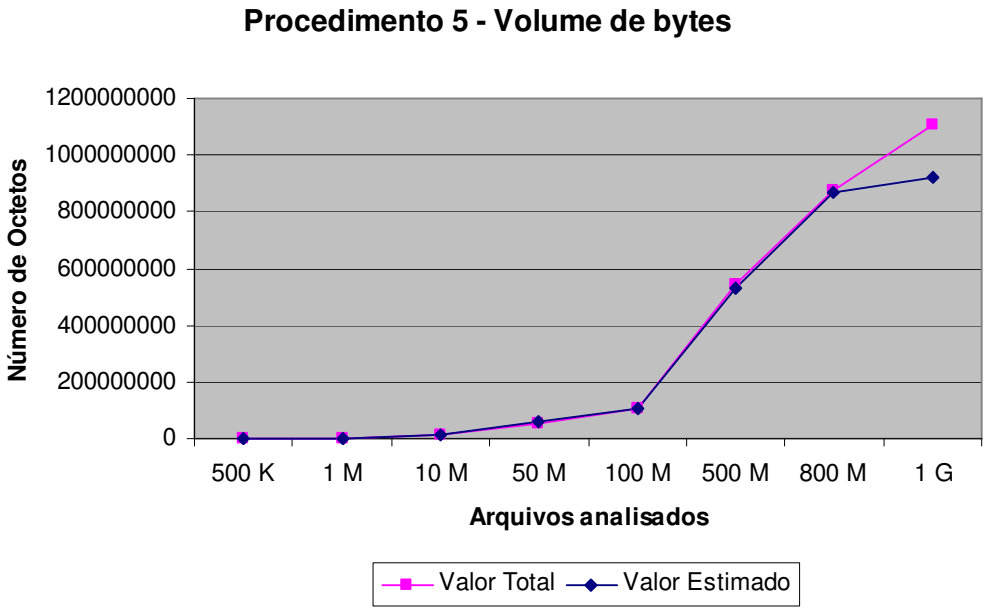



**Figura 41** Gráfico comparativo para o volume de *bytes* no procedimento 5.

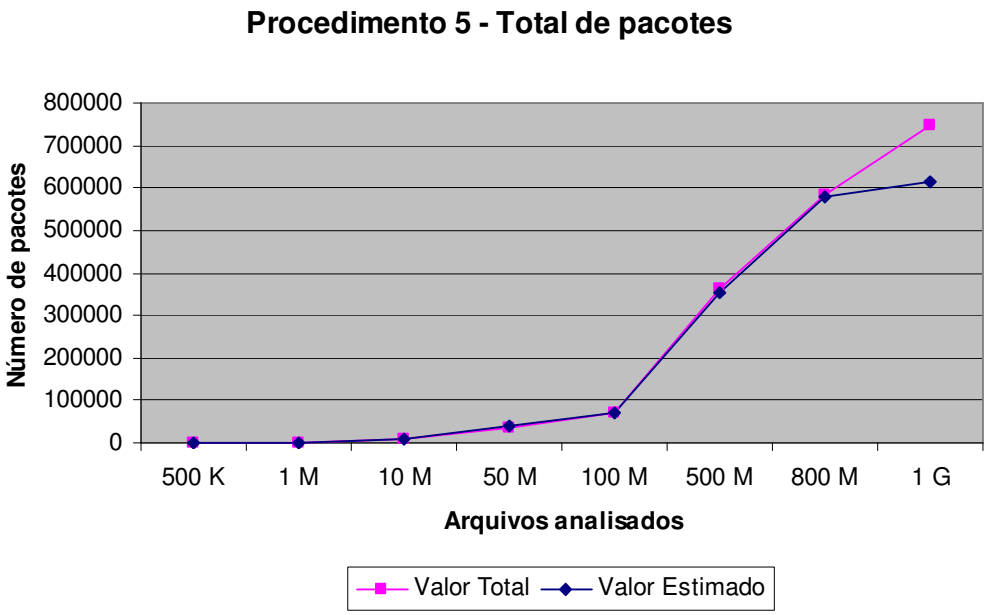



**Figura 42** Gráfico comparativo para o total de pacotes no procedimento 5.

**Tabela 36** Resultados para o volume de *bytes* no procedimento 5.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 536074 | 314092 | 41,4 |
| 1 M | 1097204 | 1003413 | 8,5 |
| 10 M | 10965633 | 12087922 | 10,2 |
| 50 M | 54788630 | 56548611 | 3,2 |
| 100 M | 109389255 | 107377832 | 1,8 |
| 500 M | 546933626 | 530362266 | 3 |
| 800 M | 874740150 | 871373261 | 0,4 |
| 1 G | 1105304924 | 919099245 | 16,8 |

**Tabela 37** Resultados para o total de pacotes no procedimento 5.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 361 | 256 | 29,1 |
| 1 M | 733 | 711 | 3 |
| 10 M | 7267 | 8033 | 10,5 |
| 50 M | 36405 | 37612 | 3,3 |
| 100 M | 72773 | 71456 | 1,8 |
| 500 M | 364478 | 353345 | 3,1 |
| 800 M | 582720 | 580645 | 0,4 |
| 1 G | 745090 | 612457 | 17,8 |

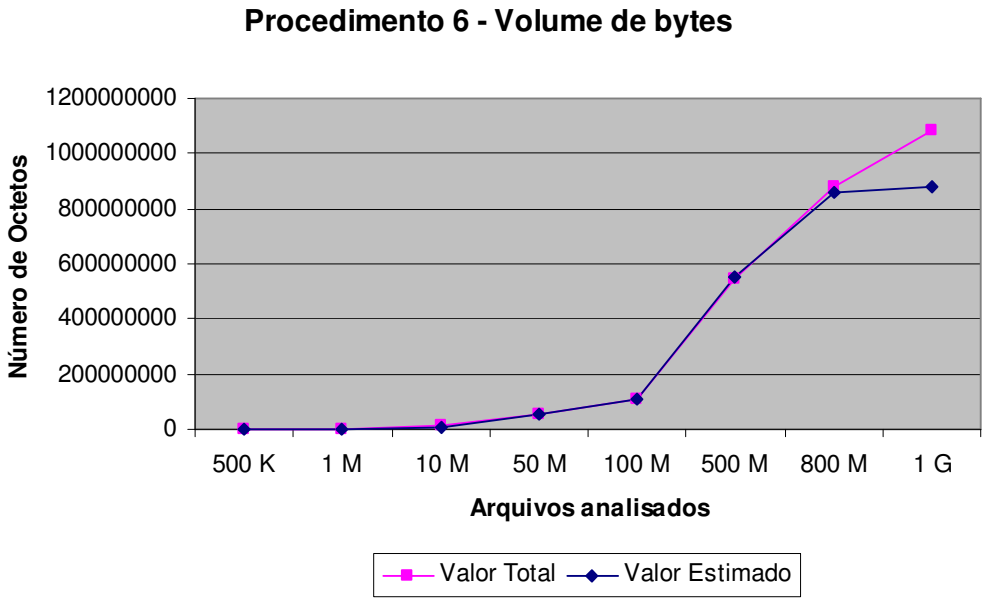



**Figura 43** Gráfico comparativo para o volume de *bytes* no procedimento 6.

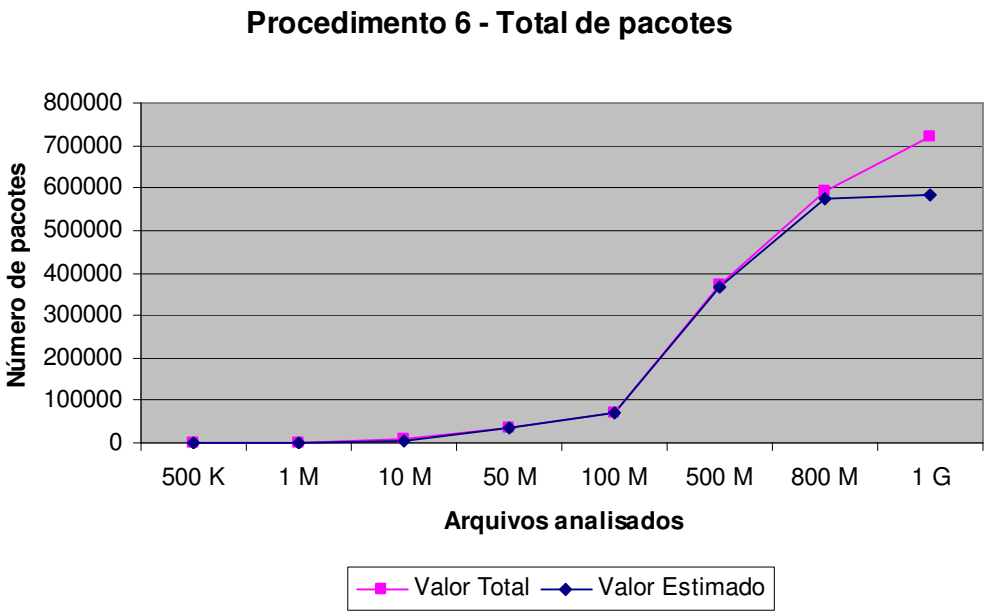



**Figura 44** Gráfico comparativo para o total de pacotes no procedimento 6.

**Tabela 38** Resultados para o volume de *bytes* no procedimento 6.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 536074 | 242775 | 54,7 |
| 1 M | 1097072 | 828034 | 24,5 |
| 10 M | 10966491 | 7840719 | 28,5 |
| 50 M | 54830594 | 54980181 | 0,3 |
| 100 M | 109659683 | 109203645 | 0,4 |
| 500 M | 548102412 | 553056134 | 0,9 |
| 800 M | 877420016 | 862393013 | 1,7 |
| 1 G | 1084627899 | 876152783 | 19,2 |

**Tabela 39** Resultados para o total de pacotes no procedimento 6.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 361 | 234 | 35,2 |
| 1 M | 731 | 620 | 15,2 |
| 10 M | 7280 | 5262 | 27,7 |
| 50 M | 36387 | 36438 | 0,1 |
| 100 M | 72755 | 72879 | 0,2 |
| 500 M | 369531 | 367423 | 0,6 |
| 800 M | 591541 | 573765 | 3 |
| 1 G | 722462 | 581656 | 19,5 |

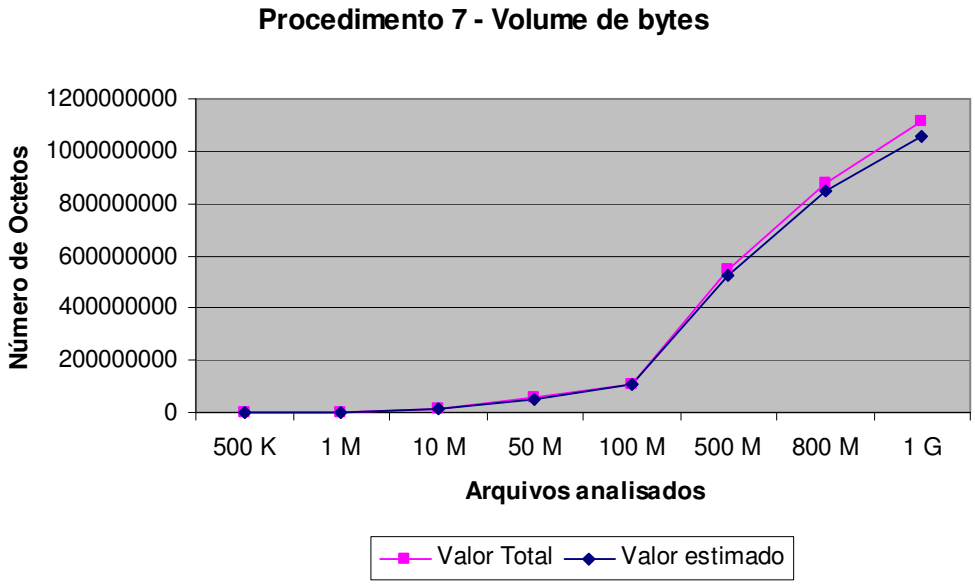



**Figura 45** Gráfico comparativo para o volume de *bytes* no procedimento 7.

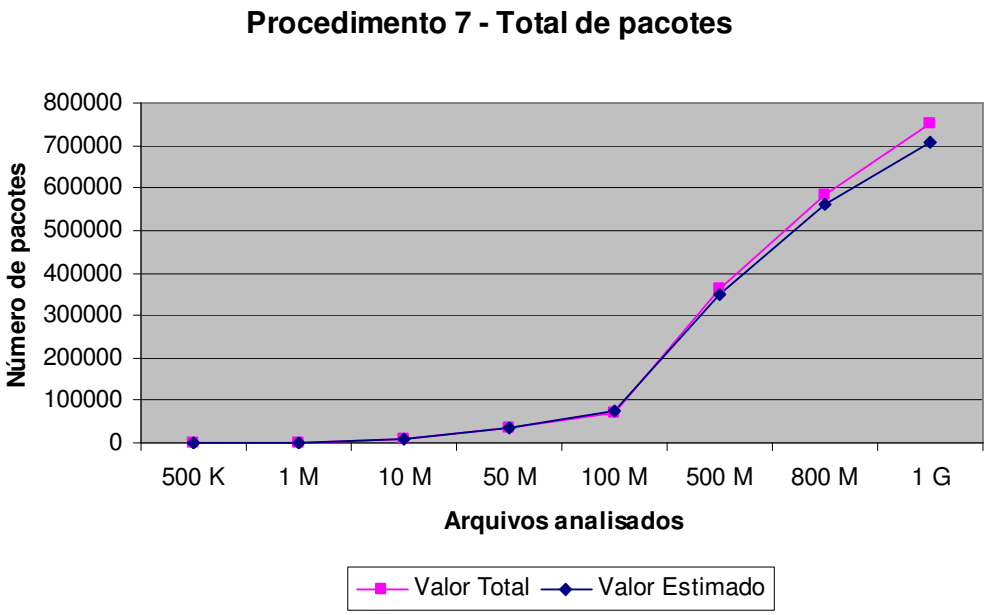



**Figura 46** Gráfico comparativo para o total de pacotes no procedimento 7.

**Tabela 40** Resultados para o volume de *bytes* no procedimento 7.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 536008 | 583596 | 8,9 |
| 1 M | 1097138 | 874113 | 20,3 |
| 10 M | 10936569 | 11378605 | 4 |
| 50 M | 54790850 | 51843398 | 5,4 |
| 100 M | 109483613 | 110813775 | 1,2 |
| 500 M | 546127962 | 523355185 | 4,2 |
| 800 M | 874694988 | 844413049 | 3,5 |
| 1 G | 1117133260 | 1058465167 | 5,3 |

**Tabela 41** Resultados para o total de pacotes no procedimento 7.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 360 | 476 | 32,2 |
| 1 M | 732 | 760 | 3,8 |
| 10 M | 7283 | 7649 | 5 |
| 50 M | 36391 | 34469 | 5,3 |
| 100 M | 72764 | 74168 | 1,9 |
| 500 M | 364010 | 349480 | 4 |
| 800 M | 583787 | 562518 | 3,6 |
| 1 G | 752746 | 708227 | 5,9 |

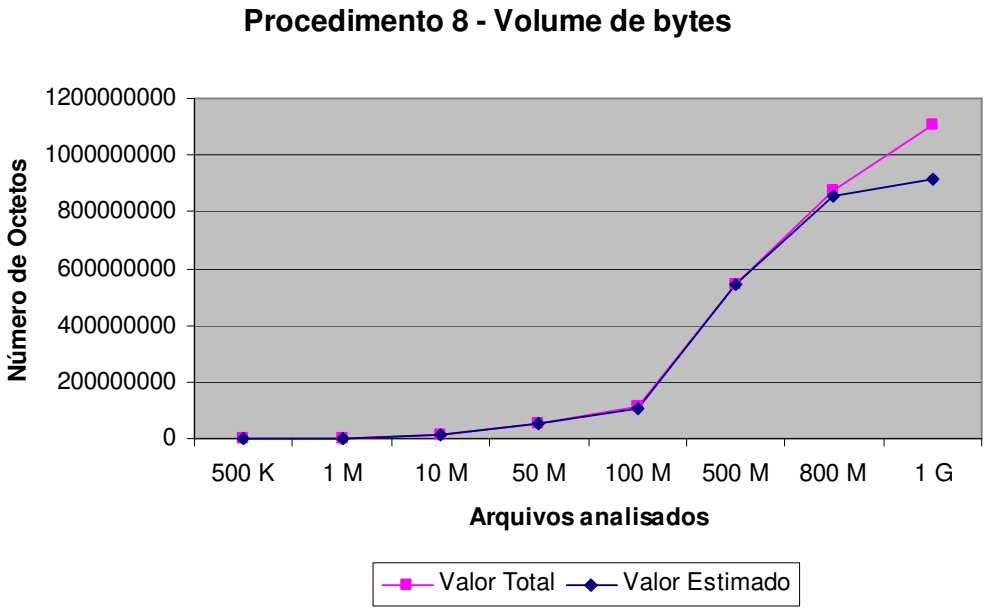



**Figura 47** Gráfico comparativo para o volume de *bytes* no procedimento 8.

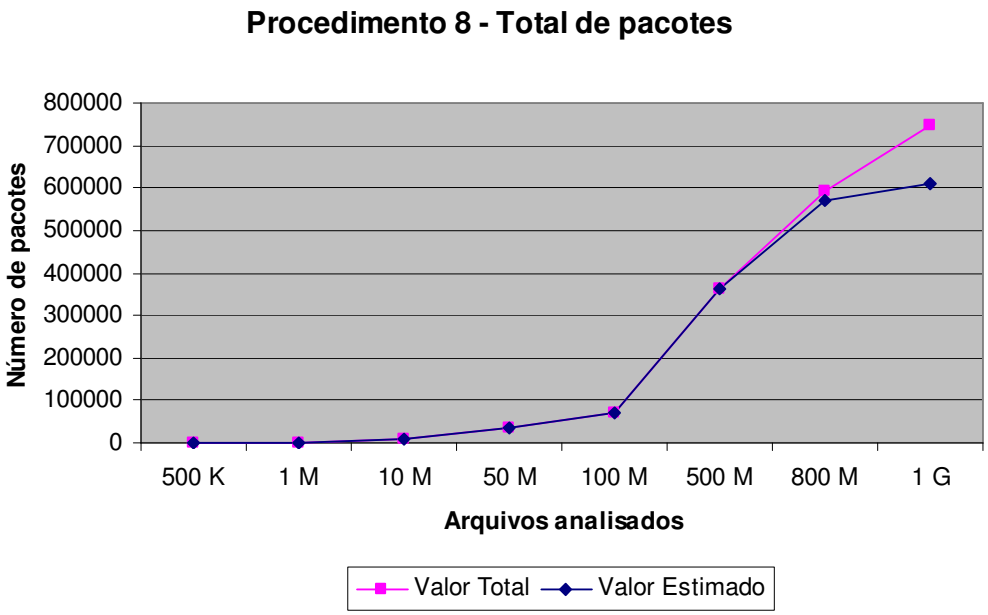



**Figura 48** Gráfico comparativo para o total de pacotes no procedimento 8.

**Tabela 42** Resultados para o volume de *bytes* no procedimento 8.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 536008 | 288447 | 46,2 |
| 1 M | 1097138 | 835440 | 23,9 |
| 10 M | 10942531 | 15330762 | 40,1 |
| 50 M | 54829736 | 50287109 | 8,3 |
| 100 M | 109665363 | 106041894 | 3,3 |
| 500 M | 546955064 | 546679585 | 0,1 |
| 800 M | 877888162 | 855333217 | 2,6 |
| 1 G | 1109606866 | 912400878 | 17,8 |

**Tabela 43** Resultados para o total de pacotes no procedimento 8.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 360 | 363 | 0,8 |
| 1 M | 732 | 724 | 1,1 |
| 10 M | 7268 | 10298 | 41,7 |
| 50 M | 36374 | 33383 | 8,2 |
| 100 M | 72863 | 70486 | 3,3 |
| 500 M | 364133 | 363183 | 0,3 |
| 800 M | 591758 | 569149 | 3,8 |
| 1 G | 747937 | 608701 | 18,6 |

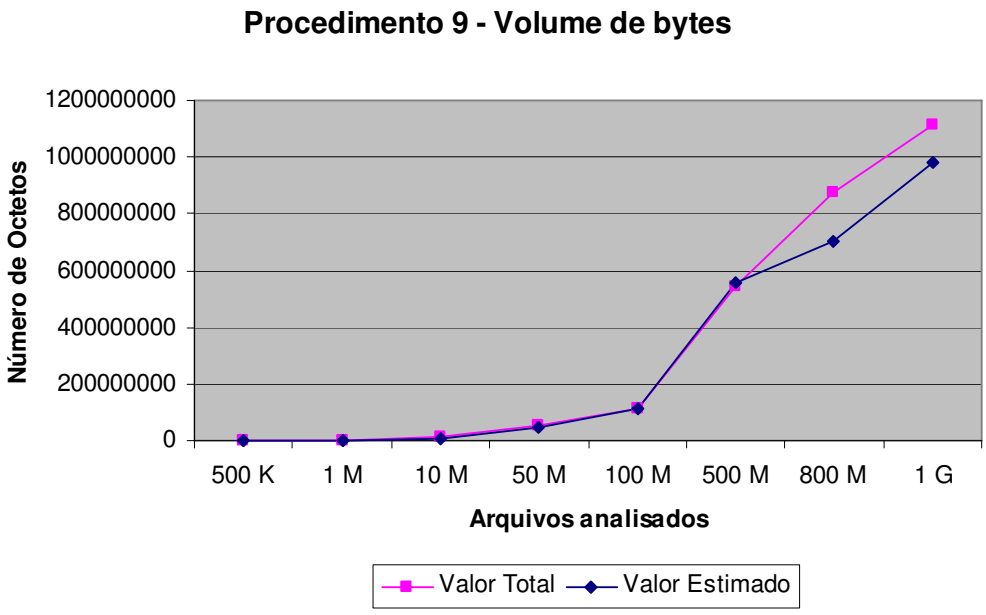



**Figura 49** Gráfico comparativo para o volume de *bytes* no procedimento 9.

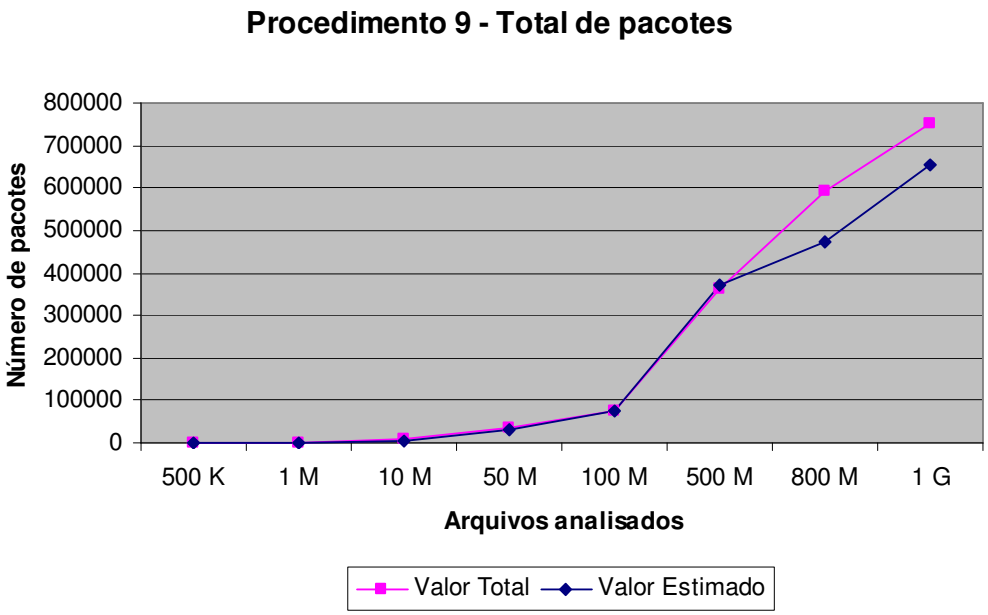



**Figura 50** Gráfico comparativo para o total de pacotes no procedimento 9.

**Tabela 44** Resultados para o volume de *bytes* no procedimento 9.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 536074 | 824641 | 53,8 |
| 1 M | 1097204 | 2834860 | 158,4 |
| 10 M | 10967349 | 8689457 | 20,8 |
| 50 M | 54800420 | 46295872 | 15,5 |
| 100 M | 109468815 | 115301584 | 5,3 |
| 500 M | 546534338 | 556190141 | 1,8 |
| 800 M | 877295000 | 703471440 | 19,8 |
| 1 G | 1113946378 | 980255928 | 12 |

**Tabela 45** Resultados para o total de pacotes no procedimento 9.

| Fluxo | Valor Total | Valor Estimado | \|Erro\| (%) |
|---|---|---|---|
| 500 K | 361 | 798 | 121,1 |
| 1 M | 733 | 2126 | 190 |
| 10 M | 7293 | 5988 | 17,9 |
| 50 M | 36452 | 31041 | 14,8 |
| 100 M | 72941 | 76807 | 5,3 |
| 500 M | 363786 | 369894 | 1,7 |
| 800 M | 591505 | 473251 | 20 |
| 1 G | 751189 | 653968 | 12,9 |

ANEXO A – Distribuição Amostral das Proporções (DAP)

Na DAP se considera uma população e todas as amostras possíveis de tamanho "*n*" que dela podem ser extraídas. Para cada amostra determina-se a proporção de sucessos relativos a um evento particular, ou seja, a probabilidade de ocorrência de sucesso. Para populações que possam ser consideradas infinitas ou para amostragem com reposição, obtém-se uma distribuição amostral onde são válidas as expressões: $\mu_p = p$ e $\sigma_P = \sqrt{\frac{p \cdot q}{n}}$ , onde $p$ é a probabilidade de ocorrência de um evento (denominado seu sucesso) e $q$ a probabilidade de não ocorrência.

Têm-se ainda que $q = 1 - p$ e, portanto o desvio padrão pode ser estimado por: $\sigma_p = \sqrt{\frac{p \cdot (1-p)}{n}}$ e a variância: $\sigma_p^2 = \frac{p \cdot (1-p)}{n}$ . Note-se que a população é distribuída binomialmente. Contudo, pelo TLC, se "*n*" for suficientemente grande ($n > 30$), a distribuição amostral é, aproximadamente, normal.

ANEXO B – Lemas para estimação do volume de *bytes*

**Lema 1:**

$\frac{n^f}{np^f} \to 1$ assim $n \to \infty$ pela lei dos grande números[7].

**Lema 2:**

Sendo $X_1, X_2, ...$ variáveis aleatórias independentes e identicamente distribuídas com média $\mu$ e variância $\hat{\sigma}$. Para cada $n$ positivo, tem-se $F_n$ como uma variável aleatória inteira positiva. Não necessita ser independente de $X_m's$. Sendo $W_n = \sum_{i=1}^{F_n} X_i$. Supondo então que $n \to \infty$, $\frac{F_n}{n}$ converge para 1, exceto quando um conjunto ou um evento de probabilidade é igual à zero. Então como $n \to \infty$, $\frac{W_n - F_n\mu}{\sigma\sqrt{n}}$ converge em uma variável aleatória com distribuição $N(0,1)$.

[7] Exceto quando um conjunto ou um evento de probabilidade é igual à zero.

ANEXO C – Implementação do Modelo Auto-Regressivo

Neste anexo é apresentado de forma simplificada, as fórmulas básicas para a implementação do modelo Auto-Regressivo AR(1). Esta explanação foi embasada a partir dos estudos relacionados à técnica de amostragem proposta por [CHO 04][CHO 06]. Basicamente, é empregada regressão linear e para estabelecimento dos parâmetros a e b previstos no modelo são aplicadas às fórmulas apresentadas abaixo.

$$m_h = a + b \cdot m_{h-1}$$

$$b = \frac{SPXY}{SQX}, onde: \begin{cases} SPXY = \sum_{j=1}^{n} \left(X_j - \overline{X}\right)\left(Y_j - \overline{Y}\right) \\ SQX = \sum_{j=1}^{n} \left(X_j - \overline{X}\right)^2 \end{cases}$$

$$a = Y_{med} - b \cdot X_{med}$$

Onde:

$m_h$ é o valor futuro (predito); *n* é o número de estados passados; *X* e *Y* são vetores de *n* elementos.

Como exemplo para implementação, pode-se supor o valor de *n = 5*. Para este caso, torna-se necessário armazenar os *n + 1* últimos valores passados, para predizer o valor futuro $m_h$. Neste contexto, os vetores *X* e *Y* são preenchidos da seguinte forma:

$$Valores\ passados = [K_1 K_2 ... K_n K_{n+1}]$$
$$X = [K_2 ... K_n K_{n+1}]$$
$$Y = [K_1 ... K_{n-1} K_n]$$

Supondo valores hipotéticos para os *n + 1* últimos valores passados iguais a {8,15,9,10,7,11} (em ordem temporal crescente, ou seja, 8 é o valor mais antigo), tem-se os vetores $X$ e $Y$, com os seguintes valores:

$$X = [15,9,10,7,11] \text{ e } Y = [8,15,9,10,7]$$

# ANEXO D – Código-fonte do inverso da função de distribuição cumulativa $\Phi(\cdot)^{-1}$

```
double PredictSampler::InverseCDF(double u) {
static double a[4] = { 2.50662823884, -18.61500062529, 41.39119773534, -25.44106049637 };
static double b[4] = { -8.47351093090, 23.08336743743, -21.06224101826, 3.13082909833 };
static double c[9] = { 0.3374754822726147, 0.9761690190917186, 0.1607979714918209,
0.0276438810333863, 0.0038405729373609, 0.0003951896511919, 0.0000321767881768,
0.0000002888167364, 0.0000003960315187 };
double x, r;
x = u - 0.5;
if (fabs(x) < 0.42) {
        r = x  * x;
        r = x  * (((a[3]*r+a[2])*r+a[1])*r+a[0])/((((b[3]*r+b[2])*r+b[1])*r+b[0])*r+1.0);
        return r;
        }
r = u;
if ( x > 0.0)
        r = 1.0 - u;
r = log(-log(r));
r = c[0]+r*(c[1]+r*(c[2]+r*(c[3]+r*(c[4]+r*(c[5]+r*(c[6]+ r*(c[7]+r*c[8])))))));
if ( x < 0.0)
        r = -r;
return r;
};
```