

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS**

RENATA MARA SCHNECKENBERG WOWK

**MONITORAÇÃO DE MÉTRICAS DE QUALIDADE DE SERVIÇO DE REDES
PARA APLICAÇÕES DE TEMPO-REAL UTILIZANDO TÉCNICAS DE
AMOSTRAGEM BASEADAS EM CONTROLE ESTATÍSTICO DE PROCESSO**

CURITIBA

2009

RENATA MARA SCHNECKENBERG WOWK

**MONITORAÇÃO DE MÉTRICAS DE QUALIDADE DE SERVIÇO DE REDES
PARA APLICAÇÕES DE TEMPO-REAL UTILIZANDO TÉCNICAS DE
AMOSTRAGEM BASEADAS EM CONTROLE ESTATÍSTICO DE PROCESSO**

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Área de Concentração: *Redes de Computadores e de Telecomunicações.*

Orientador: Prof. Dr. Edgard Jamhour.

CURITIBA

2009

RENATA MARA SCHNECKENBERG WOWK

**MONITORAÇÃO DE MÉTRICAS DE QUALIDADE DE SERVIÇO DE REDES
PARA APLICAÇÕES DE TEMPO-REAL UTILIZANDO TÉCNICAS DE
AMOSTRAGEM BASEADAS EM CONTROLE ESTATÍSTICO DE PROCESSO**

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

COMISSÃO EXAMINADORA

Prof. Dr.
Pontifícia Universidade Católica do Paraná

Prof. Dr.
Pontifícia Universidade Católica do Paraná

Prof. Dr.
Universidade Federal do Paraná

Curitiba (PR), de de 2009.

AGRADECIMENTOS

Ao professor orientador Dr. Edgard Jamhour pela orientação franca e precisa, além da paciência durante todas as reuniões para construção deste trabalho.

Ao meu esposo e família pelo constante apoio e compreensão, pelas horas de convivência que lhes foram diminuídas.

Ao Ronaldo Kleber da Fonseca e empresa Siemens que viabilizaram o curso de Mestrado e acreditaram no meu trabalho.

Aos amigos do HSBC pelo constante apoio e incentivo nos momentos difíceis.

A Deus que me deu o dom da vida, a graça da fé e a perseverança para continuar aprendendo.

RESUMO

Com a crescente implementação de serviços de rede que demandam alta qualidade, a utilização de políticas de QoS (“Quality of Service”) torna-se imprescindível para garantir tal requisito. Aplicações em tempo real como “Voz sobre IP” (VoIP) demandam monitoração de métricas de qualidade, de forma a garantir que os serviços sejam atendidos adequadamente e, em caso de problema, permitam uma rápida identificação do ponto onde a métrica não está de acordo com os níveis delineados. A proposta descrita neste documento tem o objetivo de utilizar técnicas de amostragem para monitoração de tais métricas através de um algoritmo que utiliza como base o controle estatístico de processo.

Palavras-Chave: QoS, VoIP, SLA, MDI, CEP.

ABSTRACT

With the growing implementation of network services which demand high quality services, the implementation of QoS (Quality of Service) politics is indispensable to guarantee such requirement. Applications in real time as Voice over Internet Protocol (VoIP) demand a monitoring of quality metrics in a way to guarantee that the services are being attended appropriately and in case of problem allow a fast identification where the metric is not in agreement with the required levels. The proposal described in this document has objective of using sampling techniques to monitor network metrics through an algorithm based on the statistical control of process.

Keywords: QoS, VoIP, SLA, MDI, CEP.

LISTA DE FIGURAS

Figura 2.1 Arquitetura de alto nível do modelo PSAMP	16
Figura 2.2 Comparativo entre os métodos de Seleção (ZSEBY,2005)	20
Figura 3.1 Demonstração do Cálculo de MDI.....	26
Figura 4.1 Gráfico de Controle típico.....	29
Figura 4.2 Gráfico da Distribuição Normal	31
Figura 5.1 Diagrama de blocos da metodologia utilizada	32
Figura 5.2 Topologia da rede onde foram capturados os dados.....	33
Figura 5.3 Cálculo do valor de DF	35
Figura 5.4 Gráfico com valores de DF calculados para uma coleta com 956118 pacotes de voz, utilizando o codec G729 com taxa de transmissão de 8 kbps e duração de 130 segundos.....	35
Figura 5.5 Comportamento do algoritmo de amostragem dinâmica, após três janelas mantendo o baseline, a taxa de amostragem é comutada para 30%	38
Figura 5.6 Comportamento do algoritmo de amostragem dinâmica, após duas janelas, identificado novo padrão de comportamento que resulta na criação de um novo <i>baseline</i>	39
Figura 5.7 Comportamento do algoritmo de amostragem dinâmica, após duas janelas, verificada possibilidade de um novo padrão de comportamento, assim taxa de amostragem é comutada para alta.	40
Figura 5.8 Mecanismo para verificação do funcionamento do algoritmo	41
Figura 6.1 Gráfico Original e Gráfico Obtido através do algoritmo de amostragem dinâmica para o cenário normal.....	42
Figura 6.2 Gráfico original e gráfico obtido através do algoritmo de amostragem dinâmica para Cenário Transitório	46
Figura 6.3 Gráfico original e gráfico obtido através do algoritmo de amostragem dinâmica para Cenário Recorrente	49
Figura 6.4 Gráfico original e gráfico obtido através do algoritmo de amostragem dinâmica para Cenário com mudança permanente de comportamento.....	52

LISTA DE TABELAS

Tabela 2.1 Métodos de seleção propostos pelo modelo PSAMP	18
Tabela 5.1 Descritivo de Classes de <i>QoS</i>	32
Tabela 5.2 Valores de DF para arquivo com 2682 pacotes RTP	36
Tabela 5.3 Valores de DF para arquivo com 3517 pacotes RTP	36
Tabela 5.4 Valores de DF para arquivo com 6037 pacotes RTP	36
Tabela 6.1 Detalhamento dos resultados obtidos para o Cenário Normal	44
Tabela 6.2 Detalhamento dos resultados obtidos para o Cenário Transitório.....	47
Tabela 6.3 Detalhamento dos resultados obtidos para o Cenário Recorrente	50
Tabela 6.4 Detalhamento dos resultados obtidos para o Cenário de mudança permanente de comportamento	53

LISTA DE ABREVIATURAS / GLOSSÁRIO

<i>ACL</i>	<i>Access List</i> , Lista de Acesso.
<i>CTI</i>	<i>Computer Telephony Integration</i> . Integração entre Computador e Sistema de Telefonia.
<i>DoS</i>	<i>Denial of Service</i> . Negação de serviço.
<i>IETF</i>	<i>Internet Engineering Task Force</i> . Força tarefa para engenharia da Internet.
<i>IP</i>	<i>Internet Protocol</i>
<i>IPFIX</i>	<i>IP Flow Information Export</i> , Exportação de informação de Fluxo IP.
<i>IPPM</i>	<i>IP Performance Metrics</i> . Métricas para desempenho em IP.
<i>IPTV</i>	<i>IP Televison</i> . Televisão em IP.
<i>ISO</i>	<i>International Organization for Standardization</i> . Organização Internacional para definição de padronização.
<i>LCL</i>	<i>Lower Control Limite</i> . Limite de controle inferior
<i>MDI</i>	<i>Media Delivery Index</i> . Índice de entrega de mídia.
<i>MIBs</i>	<i>Management Information Base</i> . Conjunto dos objetos Gerenciados.
<i>MPEG</i>	<i>Moving Picture Experts Group</i> . Grupo de especialistas em video.
<i>MPLS</i>	<i>Multi Protocol Label Switching</i> . Protocolo múltiplo para troca de cabeçalho.
<i>PABX</i>	<i>Private Automatic Branch Exchange</i> .
<i>PSAMP</i>	<i>Packet Sampling Protocol</i> .
<i>QoS</i>	<i>Quality of Service</i> . Qualidade de serviço.
<i>RTP</i>	<i>Real Time Protocol</i> . Protocolo em tempo real
<i>SLA</i>	<i>Service Level Agreement</i> . Acordo de Nível de Serviço.
<i>SNMP</i>	<i>Simple Network Management Protocol</i> . Protocolo para Gerenciamento de Redes.
<i>TCP</i>	<i>Transmission Control Protocol</i> . Protocolo de controle de transmissão.
<i>UCL</i>	<i>Upper Control Limit</i> . Limite de Controle Superior.
<i>UDP</i>	<i>User Datagram Protocol</i> . Protocolo de datagrama de usuário.
<i>VOIP</i>	<i>Voice over IP</i> . Voz sobre IP.

SUMÁRIO

1.	INTRODUÇÃO	10
2.	MEDIÇÕES EM REDES IP	15
2.1.	PADRÕES PROPOSTOS	15
2.1.1.	Monitoração Passiva e Ativa	15
2.1.2.	O modelo PSAMP	16
2.2.	APLICAÇÕES DAS TÉCNICAS DE MONITORAMENTO	20
3.	GRANDEZAS MONITORADAS	23
3.1.	ATRASO, VARIAÇÃO DO ATRASO, PERDA DE PACOTES	23
3.2.	MDI (<i>Media Delivery Index</i>)	25
4.	EMBASAMENTO ESTATÍSTICO	28
4.1.	CONTROLE ESTATÍSTICO DE PROCESSOS	29
4.1.1.	Construção de um gráfico de controle de processo	30
5.	PROPOSTA	32
5.1.	METODOLOGIA	32
5.2.	REDE MONITORADA	32
5.3.	ALGORITMO PARA CÁLCULO DO DELAY FACTOR (DF)	34
5.4.	ALGORITMO DE AMOSTRAGEM E CONTROLE ESTATÍSTICO DE PROCESSO	36
5.5.	VERIFICAÇÃO DO FUNCIONAMENTO DO ALGORITMO	40
6.	ESTUDO DE CASO	41
6.1.	CENÁRIO COM COMPORTAMENTO NORMAL	42
6.2.	CENÁRIO TRANSITÓRIO	46
6.3.	CENÁRIO RECORRENTE	49
6.4.	CENÁRIO COM MUDANÇA PERMANENTE DE COMPORTAMENTO	52
7.	CONCLUSÃO	55
8.	REFERÊNCIAS BIBLIOGRÁFICAS	56
9.	ANEXO A	59
10.	ANEXO B	61

1. INTRODUÇÃO

A convergência de voz e dados em uma mesma infra-estrutura de rede requer garantia de níveis de qualidade adequados para as aplicações sensíveis a atraso, como voz sobre IP (VoIP).

No momento da contratação do serviço de uma operadora de telecomunicações, um ponto importante a ser observado é a definição dos requisitos do serviço, assim como os mecanismos para medição destes requisitos e os processos que definem a aplicação de multas e recompensas quando os níveis acordados não forem atingidos e quando excederem o acordado, respectivamente.

Cada vez mais os equipamentos de rede têm mecanismos sofisticados para possibilitar a monitoração dos níveis de serviço e para realização de tal monitoração são necessárias medições de tráfego na rede envolvida. No entanto, há que se ressaltar que devido ao alto volume de tráfego existente nas redes atuais, este tipo de necessidade de monitoração pode onerar a capacidade de processamento destes equipamentos e também tornar inviável a análise destes dados.

Desta forma, a amostragem de dados torna-se um mecanismo útil para viabilizar a monitoração das métricas de qualidade em uma rede de grande porte.

1.1. MOTIVAÇÃO

O aumento no volume de dados trafegados e aumento na demanda de medições de tráfego baseadas em aplicações contribuíram de maneira significativa para a sofisticação da capacidade de medições nos equipamentos de rede. Embora estes equipamentos, tenham estas facilidades, o volume de recursos necessários para medição, armazenamento, transporte e análise dos dados coletados é muito alto.

As medições também podem ser realizadas por hardware específico para este fim, entretanto, devido à complexidade das topologias das redes atuais e o alto custo deste tipo de equipamento, esta alternativa mostra-se inviável para os administradores de rede que buscam averiguar a qualidade dos serviços que estão sendo prestados por uma operadora de telecomunicações.

Assim, ao invés de uma abordagem que considera coletas de todos os dados, propõe-se uma abordagem na qual uma parcela dos dados, isto é, uma amostragem, é utilizada para realizar inferências sobre o tráfego total.

1.2. PROPOSTA

Métrica é uma entidade que permite descrever o desempenho, a confiabilidade e o estado operacional de uma rede ou de seus elementos. As métricas típicas para monitoração de QoS são atraso (*delay*), variação do atraso (*jitter*) e perda de pacotes (*loss*) (VETTER, 2007).

O IPPM (*IP Performance Metrics*) é um grupo de trabalho do IETF (*Internet Engineering Task Force*) que tem por objetivo o desenvolvimento de padrão de métricas que possam ser aplicadas para mensurar a qualidade, desempenho e confiabilidade dos serviços de dados da Internet (PAXON *et al.*, 2008).

Algumas métricas propostas pelo IPPM já são utilizadas por plataformas como o *Surveyor* (KALIDINDI *et al.*, 1999), que é uma arquitetura para medição de performance na Internet que utiliza monitoração ativa de forma a prover informações de atraso (*delay*), perda de pacotes (*loss*) e caminho percorrido por um pacote na rede (*route*).

Um estudo do grupo de pesquisa Géant propõe a monitoração passiva destas métricas através da técnica de obtenção do campo de identificação do pacote e também do tempo em que este pacote passou pelo primeiro e segundo ponto de monitoração (NETWORK, 2006).

A proposta deste trabalho prevê o desenvolvimento de um algoritmo que, utilizando técnicas de amostragem dinâmica; que é um método de amostragem, onde a taxa de amostragem varia de acordo com o comportamento observado da rede; permite monitorar de forma eficiente o desempenho de uma rede em longos períodos de operação. A aplicação deste algoritmo pode servir como base para validação de acordos de nível de serviço (SLA).

Para aplicações em tempo real existe a demanda de uma monitoração contínua dos dados, visando garantir um menor tempo para detecção de problemas e até mesmo prever possíveis desvios de comportamento. Uma ênfase especial foi dada para monitoração de métricas que registrem o comportamento de aplicações em tempo real, pois estas precisam ser registradas continuamente durante a operação da rede.

A técnica de amostragem proposta neste trabalho pode, a princípio, ser aplicada a qualquer métrica de desempenho de rede. Contudo a ênfase foi dada na monitoração passiva da métrica denominada *Media Delivery Index - MDI*, uma nova métrica, cujo foco é o registro dos efeitos que podem afetar aplicações em tempo real, como vídeo e voz. Esta métrica é capaz de contemplar de forma simultânea o efeito do atraso, *jitter* e perda de pacotes, sendo,

portanto, bastante adequada para monitorar o desempenho de redes que transportam tráfego em tempo real (WELCH, J. & CLARCK, 2006).

1.3. OBJETIVOS GERAIS

O objetivo principal do projeto proposto é a utilização do método estatístico denominado CEP (Controle Estatístico de Processo) para registro contínuo do desempenho da rede durante longos períodos de tempo.

O controle estatístico de processo é um método efetivo de monitoração através do uso de gráficos de controle. Este método é altamente utilizado na indústria para controlar o processo de produção de um determinado produto.

Os gráficos de controle permitem o uso de um critério objetivo para distinguir uma variação de outros eventos no processo, baseando-se em técnicas estatísticas. Através da coleta de amostras em vários pontos do processo de produção, variações que possam afetar a qualidade do produto podem ser detectadas e corrigidas, reduzindo perdas e problemas que possam afetar o cliente final (MONTGOMERY, 2005).

Do nosso conhecimento, as técnicas de CEP não têm sido empregadas para o acompanhamento do desempenho de redes IP. A avaliação do seu uso para esse novo cenário é, portanto, um dos objetivos gerais desse trabalho. Na técnica de CEP original, a amostragem é constante ou total (isto, toda a população é amostrada). No nosso trabalho, contudo, dados os objetivos de redução dos custos de medição e armazenamento do processo, foi utilizado o conceito de amostragem dinâmica, onde a seleção do pacote é disparada periodicamente, de acordo com o comportamento da rede, através de um contador de pacotes, de forma que um número de pacotes é selecionado cada vez que o contador é disparado (DUFFIELD, 2007).

No nosso método, a quantidade de pacotes amostrados varia de acordo com o comportamento do desempenho da rede. Se o desempenho da rede estiver estável, a taxa de amostragem é reduzida. Quando ocorre uma variação de desempenho, a taxa de amostragem é elevada momentaneamente a fim de confirmar que se trata de uma mudança de tendência. Conforme discutido na seqüência desse trabalho, existem várias técnicas de amostragem sugeridas para redes IP. O método de seleção de pacotes escolhido neste trabalho é o de amostragem sistemática por contagem.

Os dados amostrados são utilizados para construção do gráfico de controle de processos que representa o desempenho das métricas monitoradas na rede. O gráfico CEP é

descrito em termos dos valores das amostras colhidas para cálculo da média, desvio padrão e os limites de controle que serão utilizados na monitoração. Esses valores são avaliados e, se necessário, reajustados, em intervalos regulares denominados “janelas de tempo”.

O objetivo do algoritmo é utilizar a menor taxa de amostragem possível para demonstrar um valor muito semelhante ao verdadeiro valor da métrica selecionada. A validação é feita através da verificação do comportamento da métrica através dos limites estabelecidos na construção do gráfico de controle.

Para testar o comportamento do algoritmo, de acordo com os objetivos do trabalho, foram definidos alguns cenários de testes. Estes cenários foram construídos de forma a validar o comportamento do algoritmo em diversas situações.

Para medir a eficácia do algoritmo, foi calculada para cada uma das janelas de tempo definidas, a aderência do algoritmo. O objetivo da aderência é medir a eficiência do algoritmo em perceber as transições de desempenho da métrica durante todo o período de tempo em que os dados são monitorados.

Considerando que o tráfego em uma rede de dados segue uma distribuição normal, foram estabelecidos os percentuais mínimos de pontos que deveriam estar dentro dos limites considerados de *sigma 2* e *sigma 3*. Para cada janela de tempo, os dados de pontos dentro ou fora de tais limites, foram armazenados e comparados aos do gráfico original. A aderência é calculada através da razão entre os pontos considerados dentro dos limites gráfico obtido através do algoritmo e o gráfico original.

Para cada um dos cenários montados, é calculada a aderência por janela de tempo definida, com isto, é possível validar o comportamento do algoritmo.

1.4. OBJETIVOS ESPECÍFICOS

O objetivo específico deste trabalho é aplicar o método de amostragem dinâmica para registro da grandeza denominada MDI.

O MDI é um conjunto de medidas que pode ser utilizado para monitorar qualidade da rede ou diagnosticar problemas na entrega de aplicações em tempo real como vídeo MPEG (*Moving Picture Experts Group*), que é um padrão de compressão utilizado em vídeo e áudio desenvolvido pela ISO (*International Organization for Standardization*) (MPEG, 2008).

Embora a proposta do MDI tenha ênfase no cálculo da métrica para pacotes MPEG, a metodologia pode ser utilizada em outras aplicações que sejam sensíveis a *delay* ou *jitter*, como por exemplo, VoIP (WELCH, J. & CLARCK, 2006).

O MDI é composto de dois componentes denominados DF (*Delay Factor*) e MLR (*Media Loss Rate*). O DF indica quanto tempo um conjunto de pacotes deve ser *bufferizado* para prevenir perda de pacotes, enquanto o MLR indica a taxa de pacotes perdidos ou fora de ordem em determinado intervalo de tempo.

Para cálculo do valor de DF considera-se a metodologia proposta na RFC 4445 (WELCH, J. & CLARCK, 2006). A validação do cálculo será feita utilizando uma rede real com tráfego de VoIP. Serão realizados testes em um ambiente controlado, onde várias situações de falha serão impostas na rede de forma a verificar diferentes comportamentos da métrica.

Para validação dos valores do DF calculado serão realizados estudos comparativos com uma ferramenta comercial chamada *ClearSight Network Analyzer* da empresa *ClearSight*.

1.5. ESTRUTURA DA MONOGRAFIA

Este documento está organizado da forma a seguir:

- a) o capítulo 2 apresentará os métodos para medições em redes IP, formas de monitoração e a estrutura do modelo *Packet Sampling Protocol* - PSAMP, assim como aplicações do modelo;
- b) o capítulo 3 aprofundará conceitos sobre as grandezas monitoradas como *delay*, *jitter* e MDI;
- c) o capítulo 4 discorrerá sobre o embasamento estatístico utilizado como mecanismo para validação dos valores amostrados;
- d) o capítulo 5 apresentará o algoritmo proposto e metodologia utilizada no cálculo da métrica.
- e) os estudos de caso do algoritmo proposto serão demonstrados no capítulo 6;
- f) as considerações finais e conclusões acerca dos resultados obtidos serão apresentadas no capítulo 7.

2. MEDIÇÕES EM REDES IP

Neste capítulo se fará explanações acerca das formas de monitoração de dados passiva e ativa e apresentações de alguns trabalhos já realizados que demonstram conclusões importantes a cerca de métodos a serem utilizados para validação da métrica escolhida neste trabalho. Também será apresentado neste capítulo, o modelo PSAMP (*Packet Sampling*), proposto pelo IETF, para definição de um conjunto de padrões para medições em redes IP.

2.1. PADRÕES PROPOSTOS

2.1.1. Monitoração Passiva e Ativa

Existem duas formas para monitoração de tráfego em redes IP: **monitoração passiva**, onde os dados são coletados em pontos estratégicos da rede e **monitoração ativa** que se baseia na inclusão de pacotes de teste na rede.

No método de monitoração passiva, os pacotes capturados fazem parte do tráfego existente na rede. Neste tipo de método não há sobrecarga de tráfego para teste na rede, entretanto, deve-se ter como premissa que tal tipo de monitoração pode ser somente aplicado onde o tráfego de interesse já está presente na rede, o que é o caso da maioria das aplicações, onde se deseja validar o SLA ou efetuar engenharia de tráfego (ZSEBY, ZANDER & CARLE, 2001).

No método de monitoração ativa, pacotes são injetados com o objetivo de medir certas características da rede. Medições ativas são experimentos controlados que podem ser executados em qualquer momento e em qualquer padrão de tráfego de interesse, para um objetivo específico de monitoração. Entretanto, algumas desvantagens, como a inclusão de tráfego de testes, gerando tráfego adicional e questões de segurança, devem ser levadas em conta durante a escolha do método de monitoração. Para avaliações de SLA, deve-se ter cuidado com a inserção de pacotes na rede, pois o mesmo deve ser tratado da mesma forma de um tráfego existente, para que as medições tenham precisão (ZSEBY, ZANDER & CARLE, 2001).

Um modelo composto de monitoração ativa e passiva é mostrado na arquitetura e desenvolvimento *perfSONAR* (*Performance Service Oriented Network Monitoring Architecture*), desenvolvida pelo *Joint Research Activity 1 (JRA1)* da Rede de Pesquisa GÉANT.

O *perfSONAR* tem como objetivo a monitoração de desempenho em redes de múltiplos domínios, focando na rápida resolução de problemas entre as diversas redes envolvidas. A arquitetura é composta de três diferentes camadas, a primeira responsável pela monitoração ativa e/ou passiva dos dados em vários pontos de captura, a segunda controla o recebimento e armazenamento dos dados e topologias das redes envolvidas e a terceira que possui interface de usuário (HANEMANN *et al.*, 2006).

No estudo apresentado por Hanemann *et al.* (2006), foram feitas três composições para análise de tráfego cujo caminho envolve redes distintas: agregação de tráfego em tempo, agregação de dados em espaço e concatenação em espaço. Para esta última foi apresentada uma análise de atraso (*one-way-delay*) e a validação do procedimento proposto para obtenção de desempenho em caminhos *fim-a-fim*. Esta validação possui grande importância prática, considerando que em grandes redes é impraticável a configuração de monitorações dedicadas nos vários pontos de interesse.

2.1.2. O modelo PSAMP

O PSAMP (*Packet Sampling*) é um conjunto de processos que define um método de monitoração que utiliza técnicas estatísticas de amostragem de dados. Ainda em fase de definição pelo IETF, este modelo trata do processo de captura, seleção, exportação e criação de relatório dos dados de uma rede. A motivação do PSAMP vem da necessidade de um modelo de suporte a monitoração para fins de gerenciamento de rede.

O PSAMP trata da seleção dos dados trafegados em uma rede, de forma, que estes possam representar de forma precisa e sem erros, as informações pertinentes ao estado da rede naquele dado momento em que os pacotes foram capturados.

A Figura 2.1 mostra a arquitetura do modelo PSAMP e todas as etapas suportadas pelo modelo (DUFFIELD, 2007):

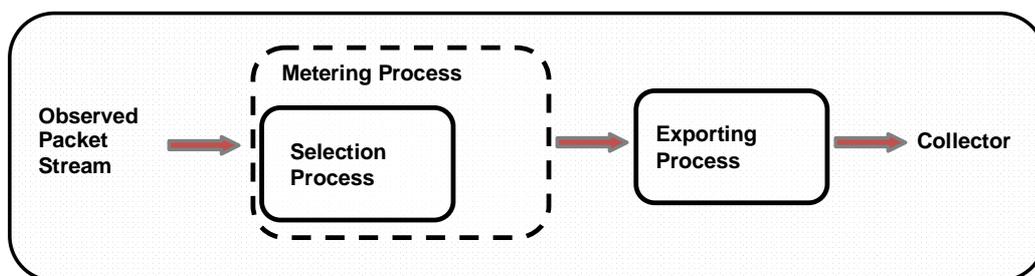


Figura 2.1 Arquitetura de alto nível do modelo PSAMP

Segue, abaixo, o detalhamento de todas as etapas do modelo PSAMP (DUFFIELD, 2007):

- a) **Conjunto de pacotes observados (*observed packet stream*):** refere-se à primeira etapa do processo, onde um conjunto de todos os pacotes observados em certo ponto da rede, denominado ponto de observação (*observation point*), podem ser visualizados. O ponto de observação pode ser uma linha na qual um dispositivo para captura está conectado, um meio compartilhado, como uma rede local, a porta de um roteador ou conjunto de interfaces físicas e lógicas de um roteador;
- b) **Processo de seleção (*selection process*):** nesta fase, o conjunto de pacotes observado passa por um processo de seleção onde um novo conjunto de dados é enviado para a próxima fase do processo. O processo de seleção é responsável pela ação de seleção, podendo ser composto ou primitivo, sendo que para o primeiro caso, um ou mais processos são executados para selecionar determinado pacote, e para o segundo caso, somente um processo de seleção ocorre. O processo de seleção pode ser categorizado de dois diferentes tipos:
 - **filtragem:** um filtro seleciona determinado pacote baseado no conteúdo, no seu tratamento ou em funções que ocorrem durante o processo de seleção;
 - **amostragem:** a seleção do pacote não pode ser determinada pelo conteúdo somente, mas deve haver algum tipo de amostragem.
- c) **Processo de contagem (*metering process*):** nesta etapa os pacotes capturados em um ou vários pontos de observação são agrupados para montagem de um relatório (report stream).
- d) **Processo de exportação (*export process*):** nesta fase, os dados do relatório montado pelo processo de contagem são enviados ao coletor de dados. O PSAMP utiliza o protocolo IPFIX (*Internet Protocol Flow Information Export*) para exportar dados de um relatório. O IPFIX, definido na RFC 3917 (QUITTEK, 2007), foi criado com o objetivo de definir um padrão para exportação de fluxo de dados do protocolo IP de roteadores, *probes* e outros equipamentos utilizados para gerenciamento de redes.
- e) **Coletor (*collector*):** recebe a informação do processo de exportação. Em alguns casos, o coletor pode ser o mesmo equipamento do processo de contagem e exportação.

As operações nos métodos de seleção são divididas em dois subgrupos (DUFFIELD, 2007):

- a) Seleção independente do conteúdo ou determinística (*content-independent sampling*): a decisão de seleção não é baseada no conteúdo dos pacotes;
- b) Seleção dependente do conteúdo (*content-dependent sampling*): na qual o conteúdo do pacote é utilizado para tomar as decisões de seleção;

A tabela 2.1 mostra os diferentes métodos de seleção suportados pelo modelo PSAMP, sua categoria e tipo de seleção (ZSEBY *et al.*, 2007):

Método de Seleção	Seleção Determinística	Dependente do Conteúdo	Categoria
Seleção Baseada em Contagem Sistemática	X	-	Amostragem
Seleção Baseada em Contador de Tempo	X	-	Amostragem
Seleção Randômica n-de-N	-	-	Amostragem
Seleção Randômica Probabilística Uniforme	-	-	Amostragem
Seleção Randômica Probabilística Não-Uniforme	-	(X)	Amostragem
Seleção Randômica Não-Uniforme por Estado de Fluxo	-	(X)	Amostragem
Filtragem por Correspondência	X	(X)	Filtragem
Seleção Baseada em Hash	X	X	Filtragem

(X) - denota os métodos de seleção que também possui independência de conteúdo.

Tabela 2.1 Métodos de seleção propostos pelo modelo PSAMP

Os métodos de seleção suportados pelo modelo PSAMP são detalhados a seguir: (ZSEBY *et al.*, 2007):

- a) **Seleção baseada em contagem sistemática (*systematic count-based*)**: a seleção é disparada periodicamente pela contagem de pacotes. O número de pacotes selecionados é determinado por uma função determinística. De acordo com Zseby (2004), este tipo de seleção demanda pouco esforço, entretanto, existe a necessidade de implementação de um contador de pacotes;
- b) **Seleção baseada em contador de tempo sistemático (*systematic time-based*)**: a seleção é disparada em instantes específicos separados por um tempo determinado. A duração dos intervalos onde são selecionados os pacotes é determinada por uma função determinística. Este tipo de seleção requer pouco esforço, não requer informações do tráfego pode ser simplesmente feita habilitando e desabilitando uma captura (ZSEBY, 2004);
- c) **Seleção probabilística n-de-N (*random n-out-of-N*)**: para a contagem de sucessivos blocos de N pacotes, n são selecionados randomicamente. Um exemplo pode ser gerar n diferentes números randômicos no *range* de $[1, N]$ e selecionar todos os pacotes que tem a posição igual ao número randômico. Este tipo de seleção requer bastante esforço devido à montagem das listas de números

randômicos. No estudo realizado por Zseby (2004) este tipo de seleção teve um resultado melhor em termos de precisão, entretanto, com um esforço muito alto e no estudo realizado por ele em 2002 (ZSEBY, 2002) o uso desta técnica não foi possível devido ao fato de que o número de pacotes em um determinado intervalo de tempo não era conhecido;

- d) **Seleção probabilística uniforme (*random uniform probabilistic*):** pacotes são selecionados de acordo com uma seleção probabilística uniforme;
- e) **Seleção probabilística não uniforme (*random non-uniform probabilistic*):** pacotes são selecionados de forma independente com probabilidade P , que depende do conteúdo do pacote. Este tipo de seleção pode ser utilizado no caso de necessidade de captura de um certo tipo de pacote que é raro, mas muito importante ex.: como no caso de informações de mudança de estado em protocolos de roteamento (ZSEBY, 2004).
- f) **Seleção randômica não-uniforme por estado do fluxo (*random non-uniform flow-state*):** este tipo de seleção é relacionado ao conceito de fluxo definido pelo na RFC3917 (QUITTEK, 2007). O pacote é selecionado de acordo com o estado do fluxo ao qual o pacote pertence. Um exemplo deste tipo de seleção é proposta por Estan e Varghese (2001), onde se um pacote pertence a fluxo que já existe no processo de armazenamento, este é selecionado e se o pacote não pertence a um fluxo já existente, ele é selecionado com a probabilidade P .
- g) **Filtragem por correspondência (*property match filtering*):** neste método, um pacote é selecionado se o conteúdo de um campo for igual a um valor pré-definido. Os campos que podem ser filtrados neste método são todos os atributos de fluxos IPFIX especificados por Quittek *et al.* (2008). Um pacote é selecionado se o valor do campo é igual ao valor determinado. Um exemplo de filtro por correspondência é a filtragem por endereçamento IP, porta TCP/UDP, interface para quais os pacotes são roteados ou tipo de mensagem de um protocolo de roteamento (ZSEBY *et al.*, 2007).
- h) **Seleção baseada em *hash* (*hash function*):** neste método de seleção, uma função de *hash* é aplicada a determinado campo do pacote capturado, se o valor cair no range definido, este será selecionado. Um exemplo de implementação deste método de seleção é mostrado no trabalho de Snoeren *et al.* (2001), no qual é apresentada uma técnica baseada em *hash* para detectar a origem de um pacote IP, entregue pela rede, em um determinado momento.

Para todos os métodos de seleção mostrados acima, deve ser considerado o esforço necessário para implementação e o nível de informação necessária de tráfego.

Os métodos probabilísticos exigem maior esforço na seleção, enquanto os sistemáticos exigem menor esforço, conforme mostrado na Figura 2.2 (ZSEBY, 2004):

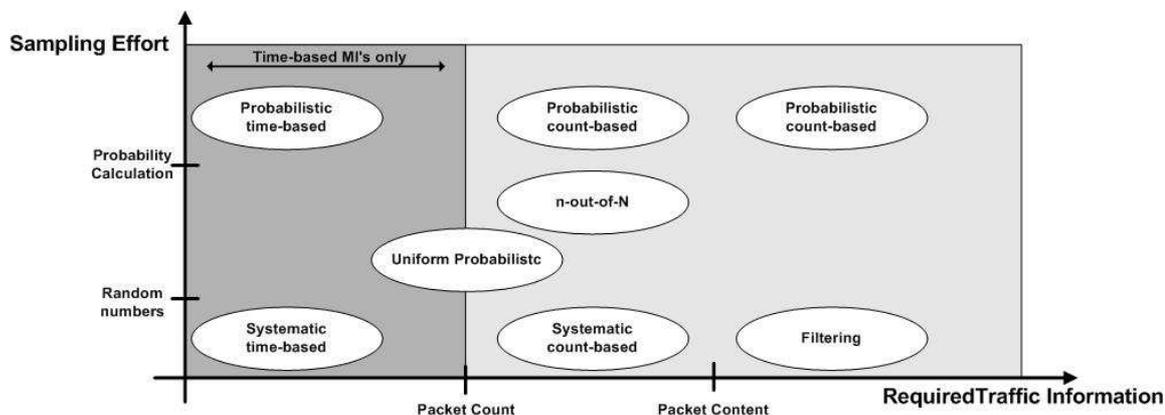


Figura 2.2 Comparativo entre os métodos de Seleção (ZSEBY,2005)

2.2. APLICAÇÕES DAS TÉCNICAS DE MONITORAMENTO

Um dos benefícios do PSAMP, conforme colocado por Zseby (2004), é o suporte a gerenciamento de novas aplicações, especificamente aquelas em que se aplicam os seletores de pacotes que o protocolo suporta.

Algumas aplicações do PSAMP são explanadas a seguir:

- a) **Medida de detalhamento (*drill down*):** a amostragem de pacotes pode ser utilizada para determinar a composição de um tráfego que atravessa uma rede. A composição pode ser realizada através de amostragens realizadas em algum ponto específico da rede e então os volumes absolutos podem ser estimados. Para realizar um detalhamento em tempo real basta criar um processo de contagem a ser disparado quando determinada característica no pacote for encontrada (DUFFIELD, 2007).
- b) **Identificação de trajetória:** o objetivo é selecionar um subconjunto de pacotes em todos os pontos de observação de uma rede e identificar a trajetória destes pacotes. A seleção de trajetória é realizada através de uma função de *hash* aplicada a uma porção do conteúdo do pacote que não varia ao longo de todo o percurso. Guan, Jian e Wei (2002) propõem um modelo de medição baseado na identificação do pacote, usando como o campo de identificação do cabeçalho IP. Usando o conceito de *entropia*, que estima o degrau de variação de um determinado processo

randômico, é comprovado que o campo de identificação não sofreu alterações durante todo o processo de transporte na rede e que houve consistência na identificação de trajetória do pacote.

- c) **Troubleshooting:** o modelo PSAMP também pode ser utilizado para diagnosticar problemas cuja ocorrência é evidente de um grupo de estatísticas, como por exemplo, utilização de determinada interface de rede ou perda de pacotes. Tais estatísticas são, geralmente, médias que foram alteradas em uma determinada janela de tempo de observação. O método mais comum para obter este tipo de informação é através da utilização da SNMP (*Simple Network Management Protocol*) MIB (*Management Information Base*). Supondo que um operador detecta que uma determinada interface de rede tem alto nível de utilização e experimenta significativos valores de perda de pacotes: este tipo de comportamento pode ter uma grande variedade de causas, dentre elas, parâmetros de roteamento incorretos, tráfego proveniente de um ataque de DoS (*denial of service*), entre outros. Na maioria dos casos, o grupo de estatísticas não é suficiente para distinguir tais causas para então decidir por uma ação corretiva. Assim, a medição de um *baseline* da interface congestionada, que pode servir para efeito comparativo para identificação do tráfego, é atípica e não auxiliar na identificação da causa da alta utilização de tráfego. Os dados da distribuição de tráfego obtido no *baseline* podem ser comparados aos valores durante o momento de alta utilização, assim, casos como DoS, podem ser identificados de forma rápida (DUFFIELD, 2007).

d) **Validação de SLA:** os métodos de amostragem podem ser utilizados para validação de valores de SLA de determinada métrica. O objetivo da validação do SLA é averiguar se os pacotes capturados estão de acordo com os níveis de serviços acordados em contrato. No trabalho de Zseby (2002), coletas não-intrusivas, em diversos pontos de uma rede, foram utilizadas para validar se os valores de *delay* estavam em conformidade com os valores estabelecidos em SLA. De acordo com este trabalho, a estimativa de média e desvio padrão de determinada métrica pode dar os primeiros indícios sobre a qualidade de uma aplicação, mas é inadequada para validar a conformidade de SLA. A informação de percentual dos valores da métrica pode prover informações sobre a situação da rede de forma geral, entretanto, se, por exemplo, 95% dos pacotes estiverem abaixo do limite estipulado para violação do SLA, não há como ter idéia do

percentual de pacotes que violou o contrato. Desta forma, ao invés de estimar o percentual dos valores da métrica, estima-se o percentual de pacotes que violaram o contrato. Para isto, pode-se utilizar a Distribuição Binomial, que é um procedimento para observar a não presença de uma determinada característica, neste caso, do percentual de pacotes que não atingiram o SLA acordado, ou seja, o percentual de violadores. Uma avaliação da qualidade da estimativa é feita através da observação da distribuição da estimativa, exemplificando, como a estimativa evolui e se foram executadas infinitas amostras. Precisam ser considerados dois importantes critérios de qualidade: a tendência, que quantifica o quanto longe da média todas as estimativas estão do valor exato e são medidas pela expectativa, e a precisão, que quantifica se as estimativas de múltiplos experimentos estão dispersas ao redor da média, e é medida pela variância. Assim, para comparação simples é utilizada a fórmula do erro padrão, utilizada para comparar os experimentos com os diversos métodos de amostragem. A conclusão obtida em Zseby (2002) demonstra que o método de seleção *n-de-N* provê a melhor exatidão com melhor esforço, a seleção probabilística requer menor esforço para coletas com muitos violadores e a amostragem sistemática mostra-se com boa exatidão e pouco esforço.

3. GRANDEZAS MONITORADAS

Neste capítulo são explanadas as grandezas comumente monitoradas para acompanhamento de desempenho de aplicações em tempo real.

Como este estudo é focado em no uso de técnicas de amostragem para validação de métricas que possam garantir a melhor experiência para os usuários de tais aplicações, algumas métricas são descritas neste capítulo e a forma de cálculo da métrica selecionada.

3.1. ATRASO, VARIAÇÃO DO ATRASO, PERDA DE PACOTES

Em redes onde aplicativos de tempo real como voz sobre IP (VoIP) e aplicações de vídeo trafegam em conjunto com as demais aplicações, mecanismos de QoS e priorização de tráfego são necessários para garantir a qualidade requerida demandada por tais aplicações. Ocorre que, independente de tais mecanismos, situações como tráfego excessivo, aumento do valor do atraso (*delay*) e até mesmo perda de pacotes podem influenciar no desempenho destas aplicações.

A monitoração constante da rede é essencial para garantir que os valores acordados em SLA com as operadoras prestadoras de serviço de telecomunicações estão sendo cumpridos.

Entre as métricas que podem ser utilizadas para validar o comportamento de aplicações em tempo real estão o atraso (*delay*), variação do atraso (*jitter*) e uma nova métrica denominada MDI (*Media Delivery Index*).

O atraso fim-a-fim, ou *round-trip-delay*, é caracterizado como sendo o tempo entre a ocorrência do primeiro *bit* de um pacote em um determinado ponto de monitoração e a ocorrência do último *bit* do próximo pacote recebido neste mesmo ponto. Para medição do valor deste atraso, os dados dos diferentes pontos de observação são correlacionados para identificar o valor do atraso que cada um dos pacotes sofreu ao passar através dos pontos de observação.

O atraso de um único caminho, ou *one-way-delay*, é uma métrica definida pelo IPPM, que considera o cálculo do atraso para somente um sentido do tráfego. O valor do atraso é calculado através da subtração entre o tempo de saída do pacote do nó origem e tempo de recebimento deste pacote no nó de destino. Esta medida é mais eficiente do que o atraso *round-trip-delay* por diversos motivos, entre eles, a diferença de trajeto que pode ocorrer entre os pacotes que trafegam em sentidos distintos (ALMES, G; KALIDINDI, S. & ZEKAUSKAS, 1999).

Existem outros componentes que podem afetar o atraso, seja o *one-way-delay* ou *round-trip-delay*, alguns estão listados abaixo (CISCO, 2006):

- a) **Atraso de processamento:** consiste no tempo que os roteadores levam para processar o cabeçalho de um pacote. Durante o processamento de um pacote, roteadores podem verificar o nível de erro ocorrido durante a transmissão de um pacote, assim como determinar o próximo destino de um determinado pacote. O *delay* de processamento em determinados roteadores é da ordem de microssegundos ou menos. Após o processamento, o roteador direciona o pacote para uma fila, onde posteriormente outro valor de *delay* pode ser adicionado (*delay* de enfileiramento). Em roteadores que executam atividade de tradução de endereços os valores de *delay* processamento são maiores, porque além da necessidade de verificação dos pacotes, também existe a modificação de pacotes recebidos e enviados.
- b) **Atraso de enfileiramento:** é o tempo que o pacote permanece nas filas de roteamento. Quando um pacote chega a um roteador, o primeiro deve ser processado e transmitido. Um roteador por processar somente um pacote de cada vez, caso os pacotes cheguem mais rápido do que o roteador pode processá-los, o ele os coloca em uma fila, também chamada de *buffer*, até que possa encaminhá-lo. O tamanho máximo da fila é proporcional ao tamanho do *buffer*. Quanto maior a fila de pacotes aguardando para transmissão, maior é a média do tempo de espera, entretanto, é preferencialmente melhor ter um *buffer* pequeno, que pode resultar em pacotes perdidos, do que longos tempos de transmissão. No momento de congestionamento de uma rede, o atraso de enfileiramento pode ser considerado infinito quando os pacotes são descartados e assim a retransmissão causa significativo atraso de rede.
- c) **Atraso de transmissão:** consiste no tempo para enviar os *bits* de um pacote para um *link*. O atraso de transmissão é uma função do tamanho dos pacotes e é dado pela seguinte fórmula $DT = N/R$, onde DT é o atraso de transmissão, N significa o número de *bits* e R a taxa de transmissão.
- d) **Atraso de propagação:** é o tempo de propagação do sinal através do meio em que os dados estão sendo transmitidos. O atraso de propagação é a distância entre dois roteadores, dividida entre a velocidade de propagação.

A variação do atraso, ou *jitter*, pode ser definida como a medida de variação do atraso entre os sucessivos pacotes de dados que trafegam em uma rede de dados. Na

transmissão de pacotes de voz, uma elevada variação ao atraso produz uma percepção ruim para os usuários deste serviço. Técnicas como *bufferização* podem ser utilizadas para diminuir o efeito da *jitter* em serviços de voz sobre IP (CISCO, 2006). O jitter pode ser calculado através da comparação do tempo de chegada dos pacotes em um nó de destino com a diferença em amostras de tempo como medido pelo nó de origem.

A perda de pacotes (*loss*) é a taxa entre o número de pacotes considerados como perdidos no nó de destino e o número de pacotes enviado pelo nó de origem. Um pacote é considerado perdido se não atingiu o destino em um período de tempo pré-determinado. A monitoração pode ser feita através da correlação de dados do cabeçalho IP dos pacotes nos pontos de coleta de dados.

3.2. MDI (*Media Delivery Index*)

Conforme comentado anteriormente, uma nova métrica definida pelo IETF, denominada MDI (*Media Delivery Index*), provê uma medida indicativa da necessidade da quantidade de *buffer* no ponto de destino para atenuar o efeito do *jitter*, assim como indicar possível perda de pacotes (WELCH, J. & CLARCK, 2006).

O MDI permite identificar rapidamente, através da monitoração constante dos dados de aplicativos de tempo real em diversos pontos da rede e em situações de carga distintas, dispositivos ou pontos que introduzem *jitter* significativo ou perda de pacotes. Através da monitoração constante dos desvios do valor nominal de *jitter* ou da perda de pacotes que podem ser utilizados para indicar uma falha eminente ou situação de carga excessiva é possível realizar um novo planejamento de capacidade (INEOQUEST, 2005a).

A RFC que descreve o MDI (WELCH, J. & CLARCK, 2006) é focada no tráfego UDP com taxas de transmissão constantes, monitorando pacotes MPEG. Como o tráfego de VoIP tem requisitos de qualidade de serviço semelhantes ao tráfego de vídeo, a metodologia para cálculo de MDI pode ser aplicada de forma igual. O método pode ser utilizado para aplicações com taxa de transmissão constante e dinâmica, embora, para esta última, exista uma maior dificuldade para os cálculos.

O MDI provê a informação necessária para detectar todos os impedimentos causados pela rede para aplicações como VoIP e vídeo, que utilizam o protocolo UDP para transporte dos dados e os métodos de transferência de dados de forma constante (AGILENT, 2008).

O MDI é medido e atualizado ao término dos intervalos de tempo determinados, abrangendo vários pacotes. Os valores mínimos e máximos de *buffer* são capturados nestes

determinados intervalos. O tempo de medida deve variar de forma a identificar uma anomalia de rede de antecipadamente (INEOQUEST, 2005a).

Os componentes do MDI são *DF* e *MLR*, onde *DF* é o *Delay Factor* e *MLR* é *Media Loss Rate*.

O *delay factor* (*DF*) é a máxima diferença, observada ao final de um intervalo de medida pré-definido (normalmente 1 segundo), entre os dados recebidos e transmitidos. O valor de *DF* deve ser atualizado ao final de cada intervalo definido e indica quando o *stream* de dados deve ser *bufferizado* em sua taxa nominal para prevenir perda de pacotes. Este tempo também deve ser visto, como uma medida de atraso da rede que deve ser adicionada pelo *buffer*, para acomodar o *jitter* e prevenir perdas (WELCH, J. & CLARCK, 2006).

O valor de *DF* estipula o valor mínimo de *buffer* requerido para o próximo *stream* de dados. Conforme o *stream* progride a variação do valor de *DF* indica o acúmulo de pacotes ou *gaps* de pacotes. Altos valores de *DF* também indicam que uma latência maior é necessária para entregar um *stream*, devido à necessidade de se completar um *buffer* recebido antes do início do escoamento de forma a garantir a entrega dos dados no tempo correto (INEOQUEST, 2005a).

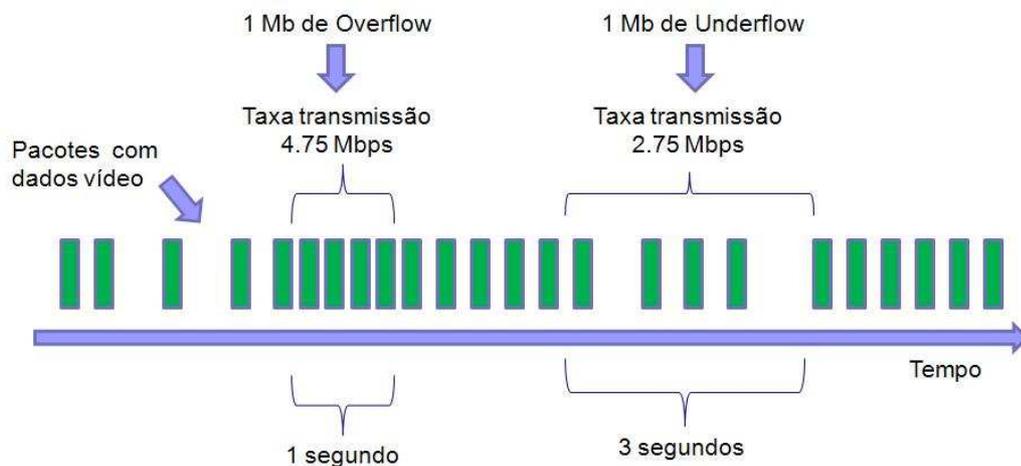


Figura 3.1 Demonstração do Cálculo de MDI

Como pode ser observado, através da Figura 3.1, considerando a transmissão de tráfego de vídeo com uma taxa de transmissão variando entre 2.75 Mbps e 4.75 Mbps, seria necessário um buffer de 1 Mb para anular o efeito do jitter. Neste caso, o valor de *DF* seria: $1 \text{ Mb} / (3.75 \text{ Mbps})$, o que resulta em 2.667ms, o que significa que o jitter adicionou 0.2667 segundos de delay nesta transmissão de dados.

Para detalhar o cálculo de DF, considere o buffer virtual (VB) usado para receber pacotes de um stream. Quando o pacote P(i) chega durante um intervalo de cálculo, dois valores de VB devem ser computados: VB(pré) e VB(pós), sendo que ambos os valores indicam a diferença de bytes recebidos e escoados. O valor do buffer virtual VB (i,pre) equivale ao tamanho buffer virtual antes da chegada de P(i) e o valor de buffer virtual VB (i,pos) equivale ao tamanho buffer virtual após chegada de P(i) (WELCH, J. & CLARCK, 2006).

$$VB(i, pre) = \sum(S_j) - MR * T_i \text{ onde } j=1 \dots i-1$$

$$VB(i, pos) = \sum(S_j) + S_i$$

onde:

S_j = tamanho do payload do pacote j

T_i = tempo relativo que o pacote i chega no intervalo

MR = taxa nominal de transferência dos dados

A condição inicial $VB(pre) = 0$ é usada no início de cada intervalo de medida.

Após obtenção dos valores de VB(pré) e VB(pós) para um cada um dos pacotes recebidos, deve ser calculado o valor de VB(Max) e VB(min), que indicam o máximo e mínimo valor de buffer virtual necessário. O valor de DF para o para o intervalo de medida é calculado da seguinte forma:

$$DF = [VB(máximo) - VB(mínimo)] / MR$$

Se não forem recebidos pacotes durante o intervalo, o último valor de DF calculado, durante o intervalo em que os pacotes chegaram, é utilizado. O tempo de chegada do último pacote é retido para utilização no cálculo de VB quando o próximo pacote chega (WELCH, J. & CLARCK, 2006).

Para o primeiro intervalo de medida, de um período selecionado para análise dos dados, nenhum valor de DF é calculado, entretanto, o tempo de chegada dos pacotes é armazenado para uso no cálculo de VB para o próximo intervalo (WELCH, J. & CLARCK, 2006).

O valor aceitável de DF pode ser considerado entre 9 e 50ms (AGILENT, 2008). O valor de DF, aceitável para uma rede específica, pode variar bastante, pois existe uma diferença no tamanho do *buffer* disponível nos equipamentos de rede.

O outro componente do MDI, denominado, MLR (*Media Loss Rate*) consiste simplesmente na contagem de pacotes perdidos ou fluxo de pacotes fora de ordem em determinado segundo. Os dados de pacotes fora de ordem são importantes, pois em muitos casos, não há reordenação de pacotes nos dispositivos de média dos clientes (AGILENT, 2008). O MLR é computado através da subtração do número de pacotes recebidos, durante o intervalo de medida definido, e o número de pacotes esperados dividido pelo tamanho do intervalo definido.

O valor máximo aceitável do MLR é próximo de zero, pois qualquer perda de pacotes poderá afetar a qualidade de voz ou vídeo. O MLR é um formato conveniente para especificar o SLA em termos de perda de pacotes. Deve ser levado em conta o componente DF, assim, um dispositivo com MDI de 4:0.0001 poderia indicar um dispositivo com fator de atraso (DF) de 4 milissegundos e com uma taxa de perda de pacotes de 0.0001 por segundo (AGILENT, 2008).

O MDI combina o fator DF (Delay Factor), que indica a possibilidade eminente de perda de dados, e o MLR (Media Loss Rate), para indicar perda ou chegada de dados fora de ordem esperada (WELCH, J. & CLARCK, 2006). O MDI é de grande importância para a localização e caracterização de problemas na rede que possam afetar a qualidade dos dados recebidos e o QoE do usuário final. Se o MDI for calculado em pontos intermediários de uma rede de grande porte, a diferença entre os valores de DF e MLR entre tais pontos pode informar rapidamente e isolar o ponto da falha (AGILENT, 2008).

4. EMBASAMENTO ESTATÍSTICO

Considerando que o comportamento de tráfego em redes de grande porte tende a variar ao longo de diferentes horários do dia e também ao longo do mês, torna-se necessário um monitoramento constante dos dados. O objetivo deste trabalho é utilizar um método de amostragem de dados que espelhe o comportamento da rede nestes diferentes períodos do dia, utilizando somente uma parcela dos dados monitorados.

Para identificação da taxa de amostragem mínima que ilustra o comportamento da métrica calculada foi utilizado o método denominado Controle Estatístico de Processo (CEP), que permite a monitoração efetiva de um processo através do uso de gráficos de controle.

O CEP é muito utilizado na indústria para controle do processo de fabricação de produtos, entretanto, em pesquisas efetuadas, a utilização do CEP para controle de desempenho em redes de dados é praticamente inexistente.

Desta forma, a adaptação do CEP para monitoramento de rede é uma importante contribuição deste trabalho.

4.1. CONTROLE ESTATÍSTICO DE PROCESSOS

Um dos principais propósitos dos gráficos de controle é rapidamente detectar uma ocorrência de mudança no processo, para que possa ocorrer uma investigação da causa da mudança e da ação corretiva que deve ser tomada.

A estrutura geral de um gráfico de controle consiste em limites de controle e uma linha de centro. Existem dois limites de controle, o limite de controle superior (UCL – Upper Control Limit) e limite de controle inferior (LCL – Lower Control Limit) (MONTGOMERY, 2005).

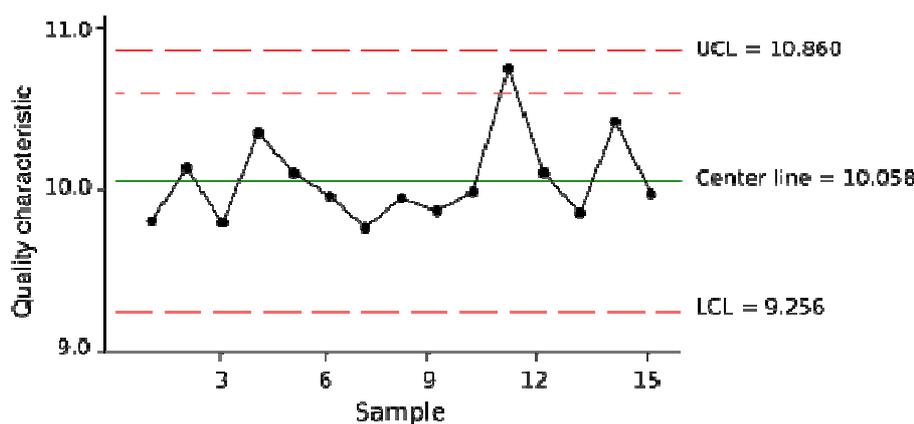


Figura 4.1 Gráfico de Controle típico

Quando um ponto de controle do gráfico fica fora dos limites de controle, considera-se que o processo está fora do controle estatístico. Existem outros padrões além deste que também indicam que um processo está fora de controle, já que é desejável que um processo esteja em controle, para que seu comportamento seja previsível.

Os gráficos de controle podem ser divididos em dois tipos:

- Gráficos de controle de variáveis:** utilizado para medição de características de valores contínuos. Este tipo de gráfico de controle é utilizado largamente, pois normalmente levam a procedimentos de controle mais eficientes e prevêm maiores informações sobre o desempenho do processo. Pode ser utilizado na área de redes para medições de métricas como atraso e variação do atraso (MONTGOMERY, 2005).

- b) **Gráficos de controle de atributos:** utilizado quando se está analisando dados como itens defeituosos em uma determinada amostra ou o número de defeitos associados a um tipo específico de item (MONTGOMERY, 2005).

Quando uma característica de qualidade é variável é uma prática comum controlar além do valor da média a variabilidade. A variação do processo ou dispersão pode ser controlada através de um gráfico de controle para médias, denominado gráfico do tipo X-barra (MONTGOMERY, 2005).

4.1.1. Construção de um gráfico de controle de processo

O gráfico de X-barra é composto por um processo com média μ e o desvio padrão σ . Supondo que o processo seja monitorado através de amostras periódicas, chamadas subgrupos de tamanho n e que a média computada para cada amostra seja \bar{X} (MONTGOMERY, 2005).

O teorema do limite central assegura que a média de cada amostra é μ , o desvio padrão da média da amostra é σ . Da mesma forma, a linha de centro para a amostra é μ e os limites inferiores e superiores são $3 \text{ sigma} * \sigma$ acima e abaixo, da linha de centro (MONTGOMERY, 2005).

$$LCL = \mu - 3 \text{ sigma} * \sigma$$

O limite de controle superior é dado pela equação:

$$UCL = \mu + 3 \text{ sigma} * \sigma$$

A média do processo é calculada através do uso da média de uma amostra periódica. Isto é dado pela equação abaixo, onde m é o número de amostras periódicas de tamanho n selecionado (MONTGOMERY, 2005).

$$\bar{\bar{X}} = \frac{\sum \bar{X}}{m}$$

A média $\bar{\bar{X}}$, também pode ser calculada somando todos os dados e então dividindo por m . O desvio padrão do processo é calculado através da raiz quadrada da variância obtida em cada intervalo de medida (MONTGOMERY, 2005).

A capacidade de um processo compara a saída de um processo em controle estatístico com a especificação dos limites utilizando os índices de capacidade (NIST, 2009).

Para executar uma análise de capacidade de um processo, o processo precisa estar em controle estatístico. É usualmente assumido que a característica de um processo sendo medido é normalmente distribuída (NIST, 2009).

A distribuição normal é considerada a distribuição de probabilidade mais importante, pois permite modelar uma infinidade de fenômenos naturais e, além disso, possibilita realizar aproximações para calcular probabilidades de muitas variáveis aleatórias que têm outras distribuições (BARBETTA, 2004). Esta distribuição é caracterizada por uma função de probabilidade cujo gráfico descreve uma curva em forma de sino, com média μ e desvio padrão σ (denominado sigma), conforme mostra a Figura 4.2:

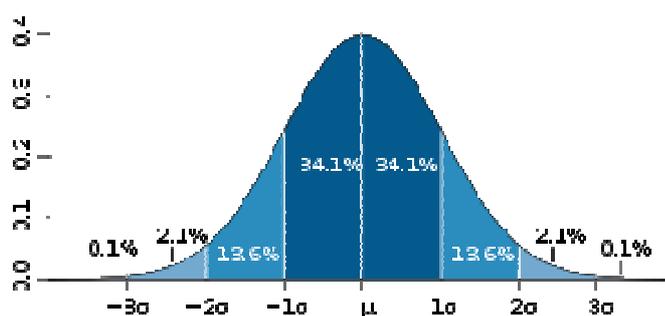


Figura 4.2 Gráfico da Distribuição Normal

Para verificar se o processo está em controle estatístico deverá ser seguida a regra descrita abaixo (BARBETTA, 2004):

- Quantidade de pontos acima dos limites de σ_2 , ou seja, a 2 vezes o desvio padrão da média μ , (denotados na Figura 4.2 como $+2\sigma$ e -2σ) deve ser no máximo 2.1% do total dos pontos;
- Quantidade de pontos acima de σ_3 , ou seja, a 3 vezes o desvio padrão da média μ , (denotados na Figura 4.2 como $+3\sigma$ e -3σ) deve ser no máximo 0.1% do total dos pontos;
- Quantidade de pontos entre média e σ_2 deve ser no mínimo 47.7% do total dos pontos;

Os dados usados para determinar se um processo está em controle estatístico podem ser usados para fazer a análise de capacidade. A distância de σ_3 em ambos os lados da média é chamado de espalhamento do processo (MONTGOMERY, 2005).

Quando as menores variações no processo devem ser monitoradas podem ser utilizados dois conjuntos de limites nos gráficos de controle. Os limites exteriores, σ_3 , são considerados os limites de *action* (ação), ou seja, quando alguns pontos estão fora deste limite deve ser verificada a causa e ação corretiva tomada. Os limites interiores, σ_2 , são considerados limites de *warning* (atenção) (MONTGOMERY, 2005).

5. PROPOSTA

5.1. METODOLOGIA

A metodologia utilizada para proposta neste documento é detalhada na Figura 5.1 e descrita a seguir:

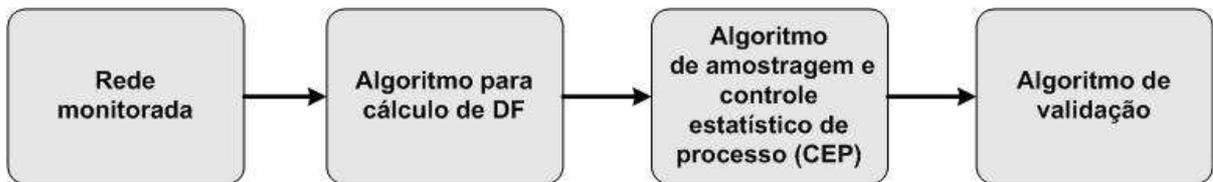


Figura 5.1 Diagrama de blocos da metodologia utilizada

5.2. REDE MONITORADA

Para validação das técnicas de amostragem serão coletados dados em uma rede real, existente em uma instituição privada.

Tal rede foi escolhida, devido ao fato de possuir subsídios para a variação de cenários requerida para validação da proposta. Alguns destes subsídios são: política de QoS definida e implementada em todos os equipamentos dos nós responsáveis pelo roteamento, possibilidade de monitoração de tráfego VoIP e acesso controlado a um ambiente de laboratório para criação dos cenários.

A política de *QoS* definida em todos os roteadores é detalhada na Tabela 5.1, na qual a aplicação de voz tem prioridade em relação às demais aplicações que trafegam na rede.

Classe	Critério	Exemplo
Prioridade 1	Aplicações onde altos valores de <i>delay</i> resultam em um nível de qualidade de serviço baixo.	VoIP, vídeo
Prioridade 2	Aplicações que demandam garantia de performance para alcançar os níveis de serviço adequados à aplicações de negócio.	Aplicações onde é necessário sincronismo de voz e dados (<i>CTI</i>)
Prioridade 3	Aplicações onde alguns valores de <i>delay</i> são tolerados se os dados forem entregues em tempos aceitáveis.	Aplicações não usadas para atendimento ao cliente externo.
Prioridade 4	Aplicações onde não existe dependência de tempo para a entrega dos dados.	Transferência de arquivos, e-mail.

Tabela 5.1 Descritivo de Classes de *QoS*

Para captura dos dados em rede, foi utilizada a ferramenta *Wireshark*, que é software livre e possibilita a captura, análise e realização de diagnósticos do tráfego de rede. O *Wireshark* provê funcionalidades muito similares ao *tcpdump* com uma interface gráfica e muito mais informações sobre opções de visualização e possibilidade de filtro de captura.

A captura de dados de chamadas VoIP foi realizada a partir de um aparelho telefônico IP conectado a um *switch* Cisco 3560. Na porta de monitoração foi espelhado o tráfego da porta do aparelho IP.

O aparelho IP é conectado a um PABX IP – Cisco Call Manager, via Rede MPLS (*Multi Protocol Label Switching*), fornecido por uma operadora de telecomunicações. A rede em questão tem 512 kbps de velocidade. O PABX IP é redundante e possui servidores distribuídos entre dois *sites* distintos, conectados via *link Gigabit Ethernet*, conforme demonstrado na Figura 5.2 abaixo:

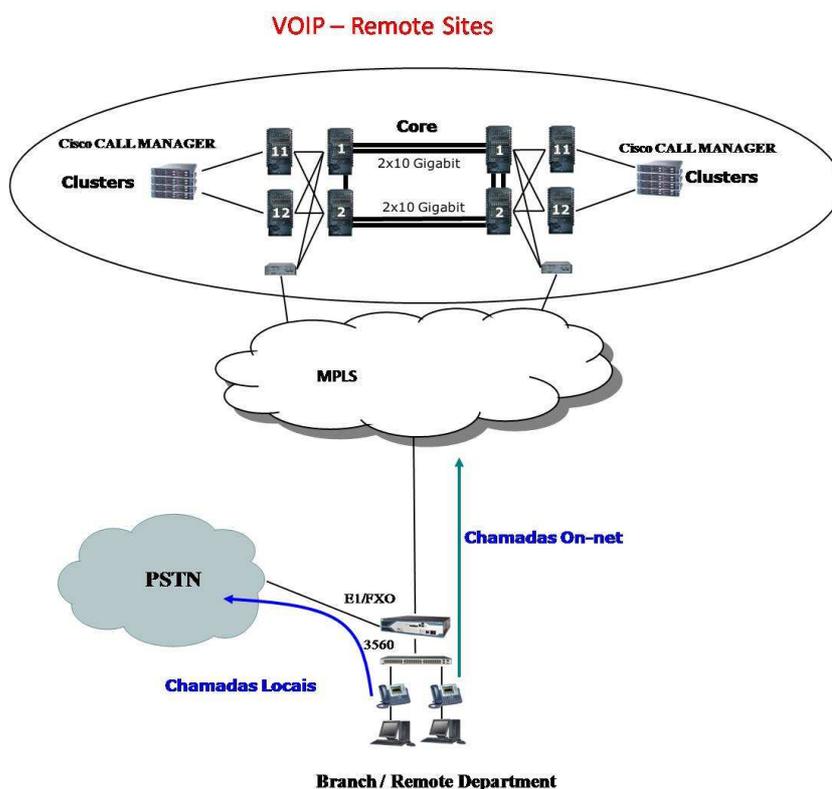


Figura 5.2 Topologia da rede onde foram capturados os dados

O *codec* utilizado para transmissão de dados de voz nas chamadas executadas para captura foi o G.729. O G.729, um algoritmo de compressão de voz que tem como requisito a baixa utilização de banda, operando originalmente em taxas de 8 *kbits/s*, podendo haver extensões que operam em até 11.9 *kbits/s*.

Foram gerados alguns cenários de tráfego, utilizando a ferramenta para geração de tráfego *iperf* e desabilitando as políticas de QoS para a classe de voz no roteador Cisco 2821. O objetivo do tráfego gerado e da retirada das políticas de QoS é fazer com que os tempos de entrega dos pacotes de voz aumente e que conseqüentemente os valores do *delay factor*

também variem. Com esta variação é possível testar o comportamento do algoritmo de amostragem dinâmica.

5.3. ALGORITMO PARA CÁLCULO DO DELAY FACTOR (DF)

A ferramenta utilizada para cálculo das métricas e simulações neste trabalho é o *R*, que é linguagem e um ambiente de desenvolvimento integrado, para cálculos estatísticos e gráficos. O *R* foi criado originalmente por Ross Ihaka e por Robert Gentleman na Universidade de Auckland, Nova Zelândia, foi desenvolvido por um esforço colaborativo de pessoas em vários locais do mundo.

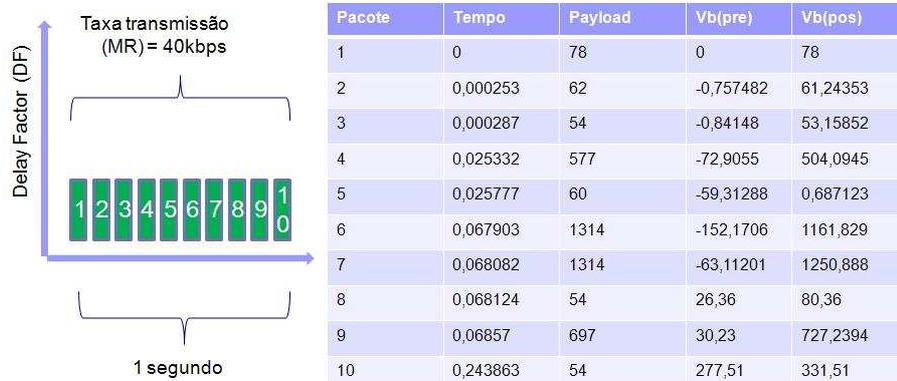
Para cálculo da métrica de DF, são necessárias informações do tamanho do *payload* e tempo relativo entre a chegada dos pacotes, desta forma, após a captura dos dados foi aplicado um filtro para seleção dos pacotes trafegados somente em um sentido (cliente1 -> cliente 2).

Após isto, foi exportado, através da ferramenta *Wireshark*, um arquivo com as informações de tempo relativo da captura do pacote na ferramenta de monitoração e tamanho de *payload* (dados) de cada pacote capturado.

Conforme recomendação da RFC 4445 (WELCH, J. & CLARCK, 2006), que estabelece um método para cálculo da métrica de *delay factor*, um valor de DF é calculado para cada segundo de captura. Desta forma, para cada pacote capturado são calculados os valores de *VBmáximo* e *VBmínimo*, e, após o término do segundo vigente, é calculado o valor do DF para este intervalo.

A Figura 5.3, demonstra o cálculo de DF para um intervalo de medida de 1 segundo, para cada pacote recebido é calculado o valor de VB (pré) e VB (pós). Ao término do intervalo, são calculados os valores de VB (máximo) e VB (mínimo), a subtração destes valores, dividida pela taxa de transmissão (MR), resulta no valor de DF para este intervalo.

Algoritmo para cálculo de DF



$$VB(pre) = \sum \text{Payload pacotes anteriores} - MR * \text{Tempo}$$

$$VB(pos) = \sum \text{Payload pacotes anteriores} + \text{Payload pacote atual}$$

$$DF = [VB(\text{máximo}) - VB(\text{mínimo})] / MR$$

Figura 5.3 Cálculo do valor de DF

A taxa nominal considerada para cálculo de DF, foi a de transmissão obtida através de detalhes da coleta, na ferramenta *Wireshark*.

Abaixo, consta um gráfico com o resultado do cálculo dos valores de DF, em diversas amostras de dados coletados:

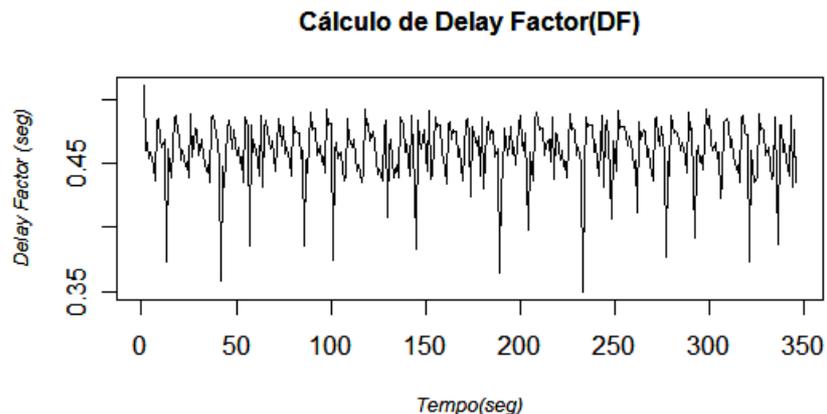


Figura 5.4 Gráfico com valores de DF calculados para uma coleta com 956118 pacotes de voz, utilizando o codec G729 com taxa de transmissão de 8 kbps e duração de 130 segundos.

A ferramenta *ClearSight Network Analyzer* da empresa *ClearSight*, é uma aplicação para monitoração e fácil identificação de problemas em rede, com suporte a IPTV e um analisador para qualidade de métricas para Voz sobre IP. Como a ferramenta também mostra os valores da métrica de DF, foram utilizados os mesmos arquivos de captura para

comparativos de valores entre a ferramenta e o *script* criado, a fim de validar o *script* criado para o cálculo do valor de DF:

Seguem abaixo, os valores encontrados durante este estudo comparativo:

Taxas Nominais	DF Médio	DF Máximo	DF Mínimo
Taxa Nominal 10,1 kips (Wireshark)	434, 7805	460, 9108	408, 6023
(% comparado ao DF calculado pelo ClearSight)	76,14%	74,10%	74,43%
Taxa Nominal 10, 894 kips (ClearSight)	466, 39784	492, 6617	437, 6339
(% comparado ao DF calculado pelo ClearSight)	81,68%	79,21%	79,71%
Valores ClearSight	571	622	549

Tabela 5.2 Valores de DF para arquivo com 2682 pacotes RTP

Taxas Nominais	DF Médio	DF Máximo	DF Mínimo
Taxa Nominal 9,8 kips (Wireshark)	413, 43945	455, 8396	64, 3016
(% comparado ao DF calculado pelo ClearSight)	70,92%	17,11%	11,71%
Taxa Nominal 10, 979 kips (ClearSight)	413, 43945	495, 76319	375, 29835
(% comparado ao DF calculado pelo ClearSight)	70,92%	18,61%	68,36%
Valores ClearSight	583	2664	549

Tabela 5.3 Valores de DF para arquivo com 3517 pacotes RTP

Taxas Nominais	DF Médio	DF Máximo	DF Mínimo
Taxa Nominal 10,2 kips (Wireshark)	432, 19204	476, 7246	317, 7884
(% comparado ao DF calculado pelo ClearSight)	75,43%	76,64%	57,88%
Taxa Nominal 10, 895 kips (ClearSight)	459, 82124	510, 215	349, 9673
(% comparado ao DF calculado pelo ClearSight)	80,25%	82,03%	63,75%
Valores ClearSight	573	622	549

Tabela 5.4 Valores de DF para arquivo com 6037 pacotes RTP

Como pode ser observado, os valores de DF médio, calculados pelo *script* desenvolvidos na ferramenta *R* são, em média, 80% do valor calculado pela ferramenta *ClearSight Analyzer*. Esta diferença é atribuída ao uso do *header* do protocolo RTP considerado como *payload* e também à diferença entre a taxa nominal calculada pelas ferramentas *Wireshark* e *ClearSight Analyzer*.

5.4. ALGORITMO DE AMOSTRAGEM E CONTROLE ESTATÍSTICO DE PROCESSO

A métrica Delay Factor (DF) é calculada de acordo com a origem e destino do tráfego *cliente -> servidor ou servidor -> cliente*, assim, para cada sentido do tráfego um valor de DF é calculado. O algoritmo criado calcula o valor de DF de acordo com a metodologia descrita na RFC 4445 (WELCH, J. & CLARCK, 2006).

Os valores da métrica DF são calculados para todos os pacotes observados. Tais valores são armazenados em um arquivo texto e o próximo passo foi a construção do algoritmo para amostragem dos dados.

Foi considerado que os primeiros 60 segundos de cada uma das coletas representam o padrão de comportamento normal do tráfego capturado, ou seja, a medida de base (*baseline*).

Assim, os valores de LCL, UCL e média são calculados utilizando os valores de DF nos primeiros 60 segundos de cada coleta, com taxa de amostragem de 100%.

Após esta primeira janela de tempo, os valores de DF começam a ser amostrados, utilizando as taxas de amostragem definidas.

As variáveis listadas abaixo, foram as definidas na construção do algoritmo de amostragem, são parametrizáveis e podem ter seus valores alterados diretamente no algoritmo:

- a) *janela*: janela de tempo considerada para análise dos dados. Foi definido para criação dos cenários com valor de 60 segundos;
- b) *sigma2*: limite de aviso a ser utilizado para construção do gráfico de controle, ou seja, $\pm 2\sigma$ (2 desvios padrão da média μ);
- c) *sigma3*: limite de ação a ser utilizado para construção do gráfico de controle, ou seja, $\pm 3\sigma$ (3 desvios padrão da média μ);
- d) *porcentagemDeAmostragemAlta*: porcentagem de dados a ser considerado como amostragem alta. Foi definido para a criação dos cenários com valor de 60%;
- e) *porcentagemDeAmostragemBaixa*: porcentagem de dados a ser considerado como amostragem baixa. Foi definido no algoritmo com valor fixo de 30%;
- f) *limiteMaxPercentualFora*: porcentagem máxima a ser considerada fora dos limites de *sigma3*. De acordo com a distribuição normal, foi definido com $0.1\% \pm 10\%$. Para seguir a distribuição normal, os pontos fora dos limites de *sigma3* devem ser respectivamente 0.1% do total de pontos. O valor de $\pm 10\%$ foi definido de forma a evitar que pequenas distorções do valor da métrica resultem em nova definição de *baseline*;
- g) *limiteMinPercentualMeio*: porcentagem mínima a ser considerada entre os limites de *sigma2* e *sigma3*. Para seguir a distribuição normal, foi definido com $2.1\% \pm 10\%$, ou seja, os pontos entre os limites de *sigma2* e *sigma3*. O valor de $\pm 10\%$ foi definido de forma a evitar que pequenas distorções do valor da métrica resultem em nova definição de *baseline*;

h) *limiteMaxPercentualDentro*: porcentagem máxima a ser considerada entre os limites de σ^2 e a média μ . De acordo com a distribuição normal, foi definido com $47.7\% \pm 10\%$. O valor de $\pm 10\%$ foi definido de forma a evitar que pequenas distorções do valor da métrica resultem em nova definição de baseline;

Para efeito de comparação foi definido o intervalo de 60 segundos, o que significa que a cada janela de 60 segundos são validados os valores da métrica. Foram definidas duas taxas de amostragem, uma considerada alta, definida como 60% e outra considerada baixa, com valor de 30%.

Os valores de LCL e UCL calculados nos primeiros 60 segundos, ou seja, durante o *baseline*, são mantidos enquanto os pontos estiverem entre os valores de sigma definidos. Se o perfil definido no *baseline* se repetir por mais de uma janela de 60 segundos, a taxa de amostragem é comutada para baixa, ou seja, 30%. Um exemplo deste comportamento é demonstrado na Figura 5.4, logo abaixo, onde após três janelas de tempo as validações foram realizadas (deve ser verificado se percentual de valores de DF mínimo, definido de acordo com a distribuição normal, está de acordo com os valores de UCL e LCL calculados na janela de *baseline*), a taxa de amostragem é comutada para 30%:

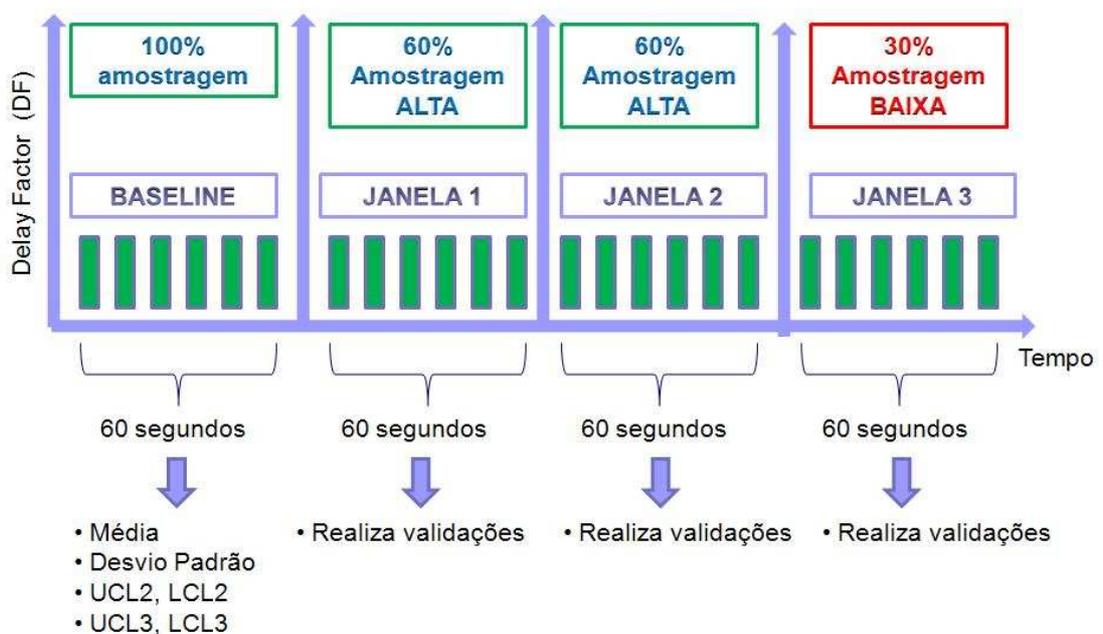


Figura 5.5 Comportamento do algoritmo de amostragem dinâmica, após três janelas mantendo o baseline, a taxa de amostragem é comutada para 30%

Um novo *baseline* é definido quando determinado número de pontos analisados estiver entre os limites da média, sigma 2 e sigma 3. O número de pontos aceitável dentro dos limites de *sigma 2* e *sigma 3* são calculados de acordo com a distribuição normal.

Abaixo, na Figura 5.5, consta comportamento do algoritmo quando necessário cálculo de um novo *baseline*, ou seja, houve uma mudança do perfil da métrica DF e a taxa de amostragem utilizada é a alta:

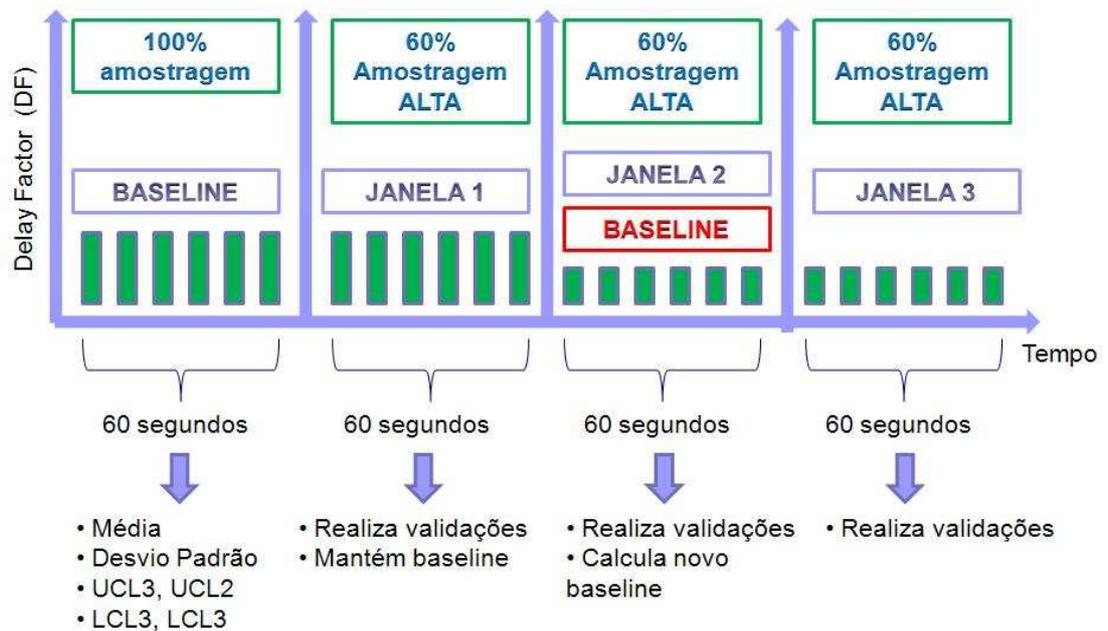


Figura 5.6 Comportamento do algoritmo de amostragem dinâmica, após duas janelas, identificado novo padrão de comportamento que resulta na criação de um novo *baseline*.

Quando houver incerteza (quantidade de pontos muito alta ou muito baixa), a taxa de amostragem é comutada para alta por mais uma janela e somente após confirmação da mudança de perfil o novo valor de *baseline* é calculado. Este comportamento do algoritmo é mostrado na Figura 5.6:

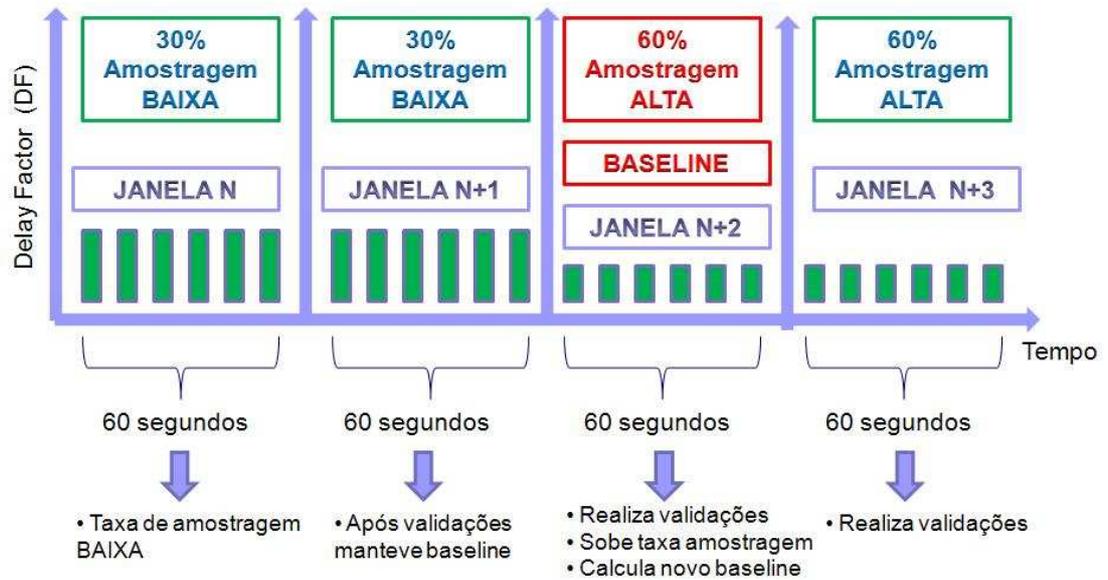


Figura 5.7 Comportamento do algoritmo de amostragem dinâmica, após duas janelas, verificada possibilidade de um novo padrão de comportamento, assim taxa de amostragem é comutada para alta.

O modelo de amostragem utilizado é de amostragem sistemática por contagem, ou seja, a cada número de valores de DF analisados na janela de 60 segundos, um é escolhido aleatoriamente, sem repetição e, assim, sucessivamente, até atingir a taxa de amostragem escolhida para a janela, sendo a taxa baixa considerada 30% e alta de 60% do total dos valores de DF calculados.

Para cada janela de 60 segundos, são armazenados os valores dentro dos limites calculados para σ_3 e σ_2 .

O número de pontos que dispara o cálculo de um novo *baseline*, ou seja, novos valores de média, LCL e UCL dependem da taxa de amostragem. Caso a taxa de amostragem seja alta e os valores da métrica ainda estejam fora dos limites estabelecidos no *baseline*, um novo *baseline* é definido e os novos valores armazenados são criados e armazenados para efeitos comparativos.

5.5. VERIFICAÇÃO DO FUNCIONAMENTO DO ALGORITMO

Para realizar a validação do algoritmo, para cada uma das janelas de tráfego definidas foi calculada a aderência do algoritmo. O algoritmo deve ter a capacidade de detectar as transições produzidas nos cenários.

O cálculo da aderência é inicialmente feito para cada uma das janelas de tempo definidas (no algoritmo a janela de tempo foi definida como 60 segundos).

O cálculo da aderência para cada uma das janelas é feito da seguinte forma:

- São verificados os pontos dentro e fora dos limites de sigma 3 que foram calculados no baseline (primeiros 60 segundos da coleta);
- É calculada a aderência desta janela, ou seja, o percentual de pontos que ficou dentro dos limites de sigma3;
- Ao final do processamento de todas as janelas é calculada a aderência total, que é a média da aderência de cada uma das janelas.

Na Figura 5.7 é mostrado o mecanismo para verificação do algoritmo de amostragem, onde para cada uma das janelas de tempo é calculado o valor da aderência:

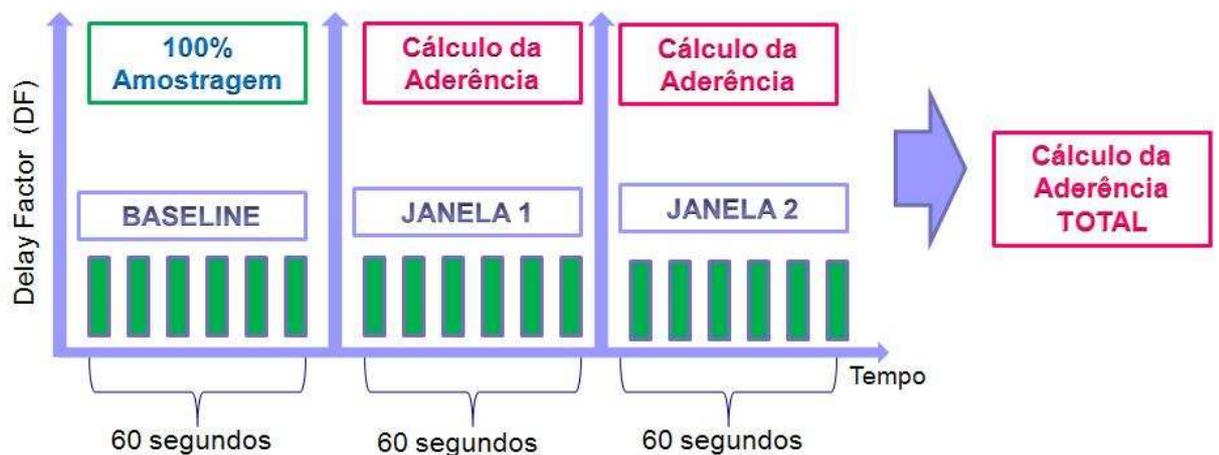


Figura 5.8 Mecanismo para verificação do funcionamento do algoritmo

6. ESTUDO DE CASO

Neste capítulo são explanados os cenários utilizados para validação do algoritmo e a análise dos resultados obtidos.

Foram definidos quatro cenários distintos com quatro coletas de tráfego VoIP distintas. Para cada coleta foram calculados valores da métrica DF.

A validação dos resultados obtidos no algoritmo foi feita através do cálculo da aderência do gráfico original e da aderência do gráfico obtido através do algoritmo, conforme metodologia explicada anteriormente.

6.1. CENÁRIO COM COMPORTAMENTO NORMAL

Neste cenário foram capturados pacotes de uma chamada de voz com nível de qualidade dentro do esperado, sem variação de tráfego ou mudança nos mecanismos de controle da qualidade do serviço.

O gráfico original e o obtido com o algoritmo para o cenário de comportamento normal demonstram de forma clara o experimento:

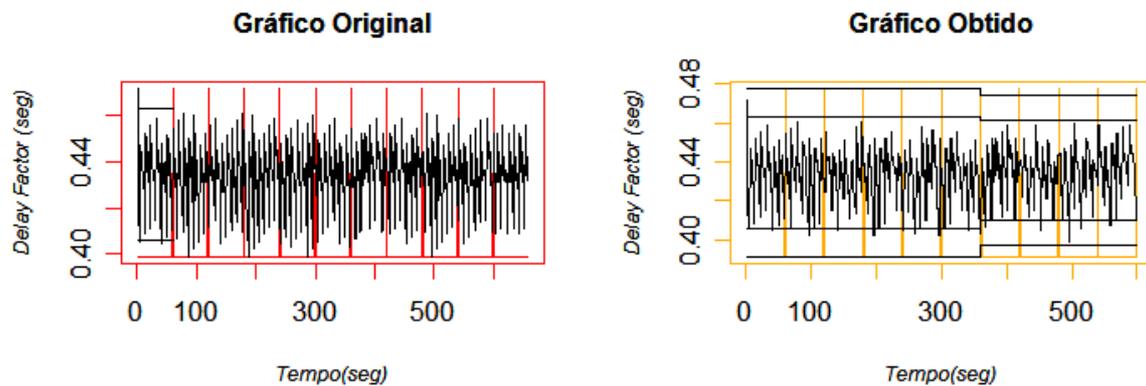


Figura 6.1 Gráfico Original e Gráfico Obtido através do algoritmo de amostragem dinâmica para o cenário normal

Na Figura 6.1, o Gráfico Original (esquerda) mostra os dados do gráfico original, isto é, o gráfico com os valores de DF calculados, já o Gráfico Obtido (direita), demonstra o gráfico obtido através do algoritmo de amostragem dinâmica.

O eixo x representa o tempo da coleta em segundos, cada uma das linhas verticais que cortam este eixo, representa a janela de tempo definida de 60 segundos, para cada janela de tempo o algoritmo executa a amostragem dinâmica e realiza as validações do *baseline*.

O eixo y representa o valor da métrica de *delay factor* (DF), calculada para cada segundo, conforme algoritmo para cálculo descrito anteriormente.

A coleta demonstrada na Figura 6.1 tem tempo total de 600 segundos, compreendendo 10 janelas de 60 segundos.

No gráfico original (intervalo de tempo de 0 a 60 segundos) são visualizadas duas linhas horizontais, que representam os valores de LCL (Limite de controle inferior) e UCL (Limite de controle superior), para σ_2 e σ_3 respectivamente. Estes valores calculados na primeira janela do gráfico original formam o *baseline*.

No gráfico obtido, são observadas quatro linhas horizontais. Essas linhas correspondem aos valores de LCL e UCL para cada uma das janelas. As duas linhas externas

representam o valor de UCL e LCL para σ_3 e as duas linhas internas, representam o valor de UCL e LCL para σ_2 .

Para este cenário, conforme pode ser verificado na Figura 6.1, no gráfico obtido não houve variação da métrica nos primeiros 350 segundos da coleta total. Após este intervalo, uma pequena variação no valor de UCL e LCL ocorreu, esta mudança pode ser percebida pela alteração das linhas de UCL e LCL para σ_2 e σ_3 e ocorreu em razão de que, conforme metodologia definida, quando ocorre certo número de pontos fora dos limites do gráfico de controle pode ocorrer aumento da taxa de amostragem ou recálculo do *baseline*.

Na tabela a seguir, constam os resultados deste cenário, na qual cada uma das linhas representa uma janela de tempo analisada e cada uma das colunas representa a seguinte informação:

- a) **Janela:** número da janela que está sendo utilizada;
- b) **Tempo Inicial:** tempo inicial considerado para de análise da janela. Ex.: para a primeira janela, o tempo inicial é 0; para a segunda janela, o tempo inicial é 61;
- c) **Tempo Final:** tempo final considerado para de análise da janela. Ex.: para a primeira janela, o tempo final é 60; para a segunda janela, o tempo final é 120;
- d) **UCL3:** representa o valor de limite superior para σ igual a 3;
- e) **UCL2:** representa o valor de limite superior para σ igual a 2;
- f) **LCL2:** representa o valor de limite inferior para σ igual a 2;
- g) **LCL3:** representa o valor de limite inferior para σ igual 3.
- h) **Taxa de amostragem:** significa o valor da taxa de amostragem utilizada no algoritmo para cada uma das janelas. As taxas de amostragem definidas no algoritmo foram definidas como baixa e alta, tendo os valores de 30% e 60%, respectivamente;
- i) **Quantidade de amostras:** quantidade de amostras utilizadas para análise. A quantidade de amostras depende do valor da taxa de amostragem, isto é, quanto mais alta a taxa de amostragem, maior o número de amostras.
- j) **Quantidade de amostras acima σ_3 :** esta coluna representa a quantidade de amostras (pontos) que ficaram acima dos limites de UCL e LCL para σ igual a 3;
- k) **Quantidade de amostras entre σ_2 e 3:** esta coluna representa a quantidade de amostras (pontos) que ficaram entre os limites de UCL e LCL entre σ_2 e σ_3 , isto é, entre UCL para σ_3 e UCL para σ_2 e LCL para σ_3 e LCL para σ_2 ;

- 1) **Quantidade de amostras entre média e σ 2:** representa a quantidade de amostras (pontos) que ficaram entre os limites da média e σ 2, isto é, entre a média e UCL e LCL para σ igual a 2.

A Tabela 6.1 demonstra os resultados do algoritmo de amostragem dinâmica para cada uma das janelas analisadas:

Janela	Tempo Inicial	Tempo Final	UCL 3	UCL 2	LCL 2	LCL 3	Taxa de Amostragem	Qtde Amostras	Qtde Amostras fora UCL3 e LCL3	Qtde Amostras entre UCL2 e LCL2 e UCL3 e LCL3	Qtde Amostras entre média e UCL2 e LCL2
0	0	60	0.4742154	0.4606979	0.4066279	0.3931104	Calculando Baseline				
1	61	120	0.4742154	0.4606979	0.4066279	0.3931104	60%	36	0	3	33
2	121	180	0.4742154	0.4606979	0.4066279	0.3931104	60%	36	0	2	34
3	181	240	0.4742154	0.4606979	0.4066279	0.3931104	60%	36	0	3	33
4	241	300	0.4742154	0.4606979	0.4066279	0.3931104	30%	18	0	1	18
4	241	300	0.4742154	0.4606979	0.4066279	0.3931104	30%	18	0	2	16
5	301	360	0.4742154	0.4606979	0.4066279	0.3931104	30%	18	0	2	16
6	361	420	0.4742154	0.4606979	0.4066279	0.3931104	30%	18	0	4	15
7	421	480	0.4742154	0.4606979	0.4066279	0.3931104	60%	36	0	4	33
8	421	480	0.4742206	0.4609033	0.4076345	0.3943172	Recalculo do baseline				
9	481	540	0.4742206	0.4609033	0.4076345	0.3943172	60%	36	0	2	34
10	541	600	0.4742206	0.4609033	0.4076345	0.3943172	60%	36	0	3	33

Tabela 6.1 Detalhamento dos resultados obtidos para o Cenário Normal

Conforme pode ser visualizado na Tabela 6.1, o algoritmo de amostragem dinâmica teve os seguintes resultados:

- Janela 0 (0-60 segundos): valores de UCL e LCL para σ 2 e 3 são calculados e armazenados como *baseline*.
- Janela 1 (61-120 segundos): os valores de UCL e LCL para σ 2 e 3 são mantidos para verificação das amostras dentro e fora dos limites estabelecidos. A taxa de amostragem inicial é de 60%. Nesta janela houve três pontos entre os limites de σ 2 e 3, como os valores mínimos entre estes limites foram atingidos não haverá cálculo de novo *baseline* e desta forma, os valores de UCL e LCL serão mantidos para análise da próxima janela;
- Janela 2 (121-180 segundos): os valores de UCL e LCL para σ 2 e 3 são os mesmos da janela anterior, a taxa de amostragem é de 60%. Nesta janela houve dois pontos entre os limites de σ 2 e 3 e, como aconteceu na janela 1, os valores de UCL e LCL serão mantidos para análise da próxima janela;
- Janela 3 (181-240 segundos): os valores de UCL e LCL para σ 2 e 3 são os mesmos da janela anterior, a taxa de amostragem é de 60% e quantidade de amostras analisadas é 36. Nesta janela houve três pontos entre os limites de σ 2 e 3 e como aconteceu na janela 1, os valores de UCL e LCL serão mantidos para

análise da próxima janela. Como o comportamento das duas janelas anteriores foi mantido, na próxima janela haverá diminuição da taxa de amostragem;

- e) Janela 4 (241-300 segundos): os valores de UCL e LCL para σ_2 e 3 são os mesmos da janela anterior, a taxa de amostragem é de 30%, uma vez que o comportamento se manteve nas das janelas 1 e 2, a quantidade de amostras analisadas é 18. Nesta janela somente um ponto ficou entre os limites de σ_2 e 3, assim, os valores de UCL e LCL serão mantidos para a próxima janela;
- f) Janela 5 (301-360 segundos): os valores de UCL e LCL para σ_2 e 3 são os mesmos da janela anterior, a taxa de amostragem é de 30%. Nesta janela somente dois pontos ficaram entre os limites de σ_2 e 3, o que significa que não haverá cálculo de novo *baseline* e que a taxa de amostragem será mantida em 30%;
- g) Janela 6 (361-420 segundos): os valores de UCL e LCL são mantidos conforme janela anterior, taxa de amostragem em 30%, quantidade de amostras analisadas é 18. Nesta janela, quatro pontos ficaram entre os limites de σ_2 e 3, este número indica que a taxa de amostragem deverá ser aumentada na próxima janela, pois, os valores máximo de ponto aceitáveis entre estes limites foi excedido;
- h) Janela 7 (421-480 segundos): valores de UCL e LCL foram mantidos conforme janela anterior, taxa de amostragem subiu para 60%, quantidade de amostras analisadas é 36. Nesta janela, quatro pontos ficaram fora dos limites estabelecidos, desta forma, um novo *baseline* será calculado;
- i) Janela 8 (481-540 segundos): valores de UCL e LCL foram recalculados devido aos limites terem sido excedidos na janela anterior. Taxa de amostragem será de 60% com 36 amostras para análise. A quantidade de pontos que ficaram entre os limites de σ_2 e média é de dois pontos, com isto, valores de UCL e LCL serão mantidos para próxima janela;
- j) Janela 9 (541-600 segundos): valores de UCL e LCL foram mantidos conforme janela anterior, taxa de amostragem permaneceu em 60%. A quantidade de pontos que ficaram entre os limites de σ_2 e média é de três pontos.

A aderência do gráfico original calculada é de 100%, uma vez que não existem variações nos valores de DF. Neste cenário a aderência calculada para o gráfico obtido através do algoritmo de amostragem dinâmica é de 90%.

6.2. CENÁRIO TRANSITÓRIO

Neste cenário foram simuladas perdas de qualidade devido ao aumento de tráfego na rede e falta de políticas de QoS adequadas para garantir a qualidade do serviço de voz.

Segue abaixo, gráfico original do cenário com comportamento transitório:

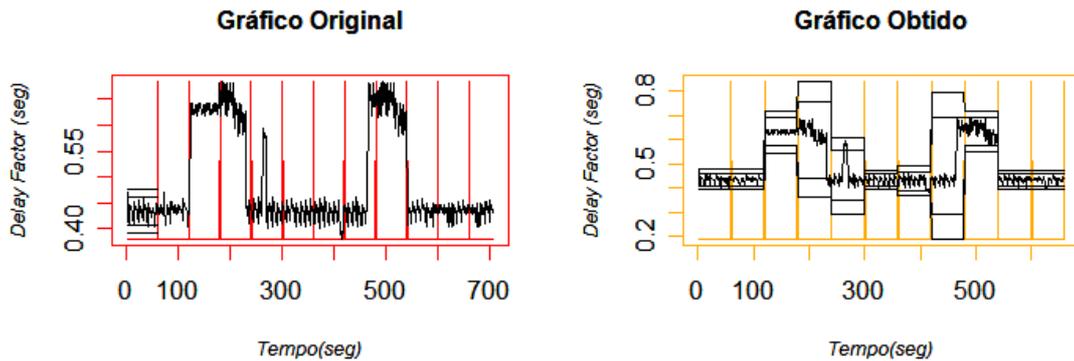


Figura 6.2 Gráfico original e gráfico obtido através do algoritmo de amostragem dinâmica para Cenário Transitório

Na Figura 6.2 pode ser observado o resultado do algoritmo para o cenário transitório: ocorrem diversas variações no valor da métrica DF, o que pode ser percebido pelas linhas de UCL e LCL no gráfico da direita (na primeira janela de tempo) que representa o gráfico obtido.

A coleta demonstrada na Figura 6.2 tem tempo total de 720 segundos, compreendendo a 12 janelas de 60 segundos.

No caso deste cenário, ocorreram vários cálculos de novo valor de *baseline*, tendo em vista a grande variação da métrica. A Tabela 6.2 mostra os resultados do algoritmo para cada uma das janelas de tempo analisadas.

Janela	Índice Inicial	Índice Final	UCL 3	UCL 2	LCL 2	LCL 3	Taxa de Amostragem	Qtde Amostras	Qtde Amostras fora UCL3 e LCL3	Qtde Amostras entre UCL2 e LCL2 e UCL3 e LCL3	Qtde Amostras entre média e UCL2 e LCL2
0	0	60	0.4757924	0.4619612	0.4066364	0.3928052	Calculando Baseline				
1	61	120	0.4757924	0.4619612	0.4066364	0.3928052	60%	36	0	2	34
2	121	180	0.4757924	0.4619612	0.4066364	0.3928052	60%	36	36	0	0
2	121	180	0.716853	0.6883107	0.5741415	0.5455992	Calculando Novo Baseline				
3	181	240	0.716853	0.6883107	0.5741415	0.5455992	60%	36	7	0	29
3	181	240	0.837371	0.7580978	0.4410052	0.3617321	Calculando Novo Baseline				
4	241	300	0.837371	0.7580978	0.4410052	0.3617321	60%	36	0	22	14
4	241	300	0.6104441	0.55779	0.3471735	0.2945194	Calculando Novo Baseline				
5	301	360	0.6104441	0.55779	0.3471735	0.2945194	60%	36	0	0	36
5	301	360	0.4729966	0.4598456	0.4072415	0.3940905	Calculando Novo Baseline				
6	361	420	0.4729966	0.4598456	0.4072415	0.3940905	60%	36	4	3	29
6	361	420	0.4896695	0.4691987	0.3873157	0.3668449	Calculando Novo Baseline				
7	421	480	0.4896695	0.4691987	0.3873157	0.3668449	60%	36	10	0	26
7	421	480	0.791703	0.6916258	0.2913173	0.1912402	Calculando Novo Baseline				
8	481	540	0.791703	0.6916258	0.2913173	0.1912402	60%	36	0	0	36

8	481	540	0.7202127	0.6916317	0.5773078	0.5487268	Calculando Novo Baseline				
9	541	600	0.7202127	0.6916317	0.5773078	0.5487268	60%	36	36	0	0
9	541	600	0.4714342	0.4591074	0.4098005	0.3974738	Calculando Novo Baseline				
10	601	660	0.4714342	0.4591074	0.4098005	0.3974738	60%	36	0	3	33
11	661	720	0.4714342	0.4591074	0.4098005	0.3974738	60%	36	0	5	31

Tabela 6.2 Detalhamento dos resultados obtidos para o Cenário Transitório

Conforme pode ser acompanhado através da Tabela 6.2, o algoritmo de amostragem dinâmica teve os seguintes resultados:

- a) Janela 0 (0-60 segundos): valores de UCL e LCL para σ_2 e 3 são calculados e armazenados como *baseline*;
- b) Janela 1 (61-120 segundos): valores de UCL e LCL mantidos, conforme janela anterior. A taxa de amostragem inicial é de 60%, enquanto o número de amostras analisadas é 36. Nesta janela houve dois pontos entre os limites de σ_2 e 3 e como os valores entre estes limites foram atingidos, não haverá recálculo de novo *baseline* ou diminuição da taxa de amostragem;
- c) Janela 2 (121-180 segundos): valores de UCL e LCL são mantidos os mesmos da janela anterior. A taxa de amostragem é de 60% e número de amostras analisadas é 36. Nesta janela nenhum ponto ficou entre os limites de σ_2 e 3, o que implica uma mudança de perfil e condição para recálculo do *baseline*.
- d) Janela 3 (181-240 segundos): valores de UCL e LCL foram recalculados devido a não existência do número de pontos mínimos entre σ_2 e 3. A taxa de amostragem é de 60% com quantidade de 36 amostras analisadas. A quantidade de pontos acima de σ_3 é de sete pontos, condição que demonstra que houve possível mudança de comportamento e desta forma, valor de *baseline* será calculado;
- e) Janela 4 (241-300 segundos): valores de UCL e LCL foram recalculados devido ao grande número de pontos acima do limite de σ_3 . A taxa de amostragem é de 60% com total de 36 amostras para análise. O número de pontos entre σ_2 e 3 é vinte e dois, o que também dispara novo cálculo de *baseline*.
- f) Janela 5 (301-360 segundos): o valor do *baseline* foi recalculado na janela anterior, e os valores de UCL e LCL foram alterados para esta janela. A taxa de amostragem é de 60% com quantidade de 36 amostras para análise. O número de pontos entre os limites de σ_2 e 3 é zero, o que também dispara um novo *baseline*.
- g) Janela 6 (361-420 segundos): os valores de *baseline* foram novamente recalculados devido ao fato de não existirem pontos entre os limites de σ_2 e 3. Esta condição foi montada de forma a identificar quando um tráfego permanece durante

um grande período de tempo com comportamento atípico e depois retorna à condição normal. Para esta janela, o número de pontos acima de σ_3 é quatro, entre σ_2 e σ_3 foram verificados três pontos. Esta condição dispara o recálculo do *baseline*;

- h) Janela 7 (421-480 segundos): os valores de *baseline* foram recalculados, conforme condição descrita na janela 6. A taxa de amostragem é de 60% e foram analisadas 36 amostras. Identificado nesta janela, a quantidade de dez pontos acima dos limites de σ_3 , o que automaticamente dispara um novo *baseline*;
- i) Janela 8 (481-540 segundos): o recálculo do valor de *baseline* na janela anterior, resultou em novos valores de UCL e LCL para ela. A taxa de amostragem é de 60%, com 36 amostras para análise. Como não houve pontos entre os limites de σ_2 e σ_3 , condição também observada na janela 5, novo cálculo de *baseline* será feito.
- j) Janela 9 (541-600 segundos): *baseline* recalculado devido a condição explanada na janela 8. Número de amostras é 36, pois a taxa de amostragem é de 60%. Novamente não houve pontos entre os limites de σ_2 e 3 e desta forma, novo *baseline* será calculado.
- k) Janela 10 (601-660 segundos): os valores de UCL e LCL sofreram alteração em relação a janela 8, devido à condição disparada na janela 9. A taxa de amostragem é de 60% com 36 amostras coletadas para análise. A quantidade de pontos entre limites de σ_2 e 3, para esta janela é três, condição que mantém o *baseline* para a próxima janela.
- l) Janela 11 (661-720 segundos): foram mantidos os mesmos valores de *baseline*, conforme condição na janela 9. Taxa de amostragem de 60% com 36 amostras selecionadas para análise. Quantidade de pontos entre σ_2 e 3 é de cinco pontos.

A aderência do gráfico original calculada é de 71%. Neste cenário a aderência calculada para o gráfico obtido através do algoritmo de amostragem dinâmica é de 98.3%.

A mudança dos valores de *baseline* em diversas janelas deste cenário, pode ser verificada no gráfico do cenário obtido na Figura 6.2, através da mudança nas linhas de UCL e LCL externas e internas. A condição verificada nas janelas 5 e 9 é uma conclusão importante deste cenário, para o caso de a quantidade de pontos mínima estabelecida para cada um dos limites não ser cumprida, um novo valor de *baseline* é calculado. Esta condição foi mapeada no algoritmo e tem objetivo identificar o caso de retorno de uma condição atípica de comportamento para típica.

6.3. CENÁRIO RECORRENTE

Neste cenário foram simuladas perdas de qualidade devido ao aumento de tráfego na rede de forma a obter um cenário recorrente, onde alterações no valor da métrica ocorrem de forma semelhante em determinados períodos de tempo, para testar a capacidade do algoritmo.

Segue abaixo, gráfico original do cenário com comportamento recorrente:

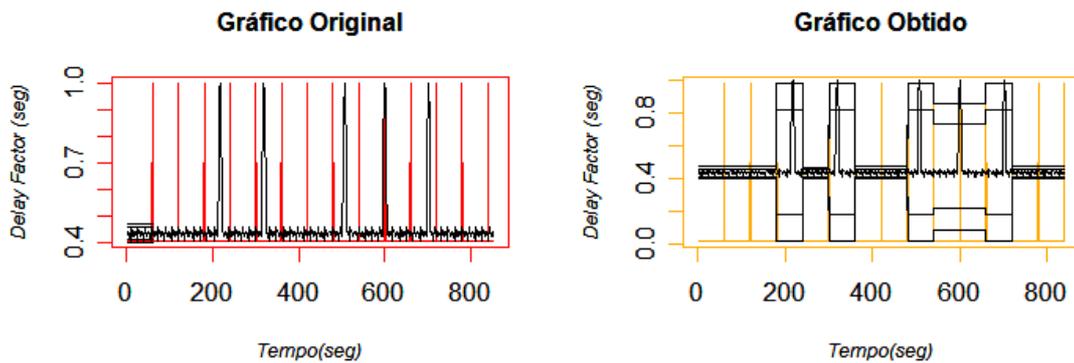


Figura 6.3 Gráfico original e gráfico obtido através do algoritmo de amostragem dinâmica para Cenário Recorrente

No cenário recorrente, o objetivo é testar a capacidade do algoritmo de reconhecer mudanças recorrentes de comportamento e voltar ao valor de *baseline* correto após as alterações de perfil.

A coleta demonstrada na Figura 6.3 tem tempo total de 720 segundos, compreendendo 12 janelas de 60 segundos.

A Tabela 6.3 mostra os resultados do algoritmo com detalhes de cada uma das janelas:

Janela	Índice Inicial	Índice Final	UCL 3	UCL 2	LCL 2	LCL 3	Taxa de Amostragem	Qtde Amostras	Qtde Amostras fora UCL3 e LCL3	Qtde Amostras entre UCL2 e LCL2 e UCL3 e LCL3	Qtde Amostras entre média e UCL2 e LCL2
0	0	60	0.4694923	0.4578048	0.4110549	0.3993674	Calculando Baseline				
1	61	120	0.4694923	0.4578048	0.4110549	0.3993674	60%	36	0	2	34
2	121	180	0.4694923	0.4578048	0.4110549	0.3993674	60%	36	0	2	34
3	181	240	0.4694923	0.4578048	0.4110549	0.3993674	60%	36	6	1	29
3	181	240	0.9764281	0.8168006	0.1782904	0.01866291	Calculando Novo Baseline				
4	241	300	0.9764281	0.8168006	0.1782904	0.01866291	60%	36	0	0	36
4	241	300	0.4686973	0.4570204	0.4103129	0.398636	Calculando Novo Baseline				
5	301	360	0.4686973	0.4570204	0.4103129	0.398636	60%	36	5	2	29
5	301	360	0.976218	0.816659	0.1784225	0.01886343	Calculando Novo Baseline				
6	361	420	0.976218	0.816659	0.1784225	0.01886343	60%	36	0	0	36
6	361	420	0.4693746	0.4575807	0.4104049	0.398611	Calculando Novo Baseline				
7	421	480	0.4693746	0.4575807	0.4104049	0.398611	60%	36	0	3	33
8	481	540	0.4693746	0.4575807	0.4104049	0.398611	60%	36	6	3	27
8	481	540	0.9764125	0.8168009	0.1783544	0.01874273	Calculando Novo Baseline				
9	541	600	0.9764125	0.8168009	0.1783544	0.01874273	60%	36	1	2	33
9	541	600	0.857636	0.7287692	0.2133018	0.0844349	Calculando Novo Baseline				
10	601	660	0.857636	0.7287692	0.2133018	0.0844349	60%	36	0	1	35
11	661	720	0.857636	0.7287692	0.2133018	0.0844349	60%	36	3	1	32

Tabela 6.3 Detalhamento dos resultados obtidos para o Cenário Recorrente

Conforme pode ser acompanhado através da Tabela 6.3, o algoritmo de amostragem dinâmica teve os seguintes resultados:

- Janela 0 (0-60 segundos): valores de UCL e LCL para σ_2 e σ_3 são calculados e armazenados como *baseline*;
- Janela 1 (61-120 segundos): valores de UCL e LCL mantido conforme definido na janela de *baseline*. A taxa de amostragem inicial de 60% tem total de 36 amostras para análise. A quantidade de pontos entre os limites de σ_2 e σ_3 é de dois pontos, condição que mantém o valor de *baseline* calculado na primeira janela;
- Janela 2 (121-180 segundos): valores de UCL e LCL mantidos. Taxa de amostragem de 60% com 36 amostras selecionadas para análise. Quantidade de pontos entre os limites de σ_2 e σ_3 é dois. Esta condição, mesmo se verificada na janela 1, mantém o valor de *baseline*;
- Janela 3 (181-240 segundos): valores de UCL e LCL mantidos, conforme janela anterior. Quantidade de amostras analisadas é 36 com taxa de amostragem em 60%. Quantidade de amostras acima de σ_3 é 6, enquanto somente um ponto foi verificado entre os limites de σ_2 e σ_3 , o que representa uma condição para recálculo do *baseline*;
- Janela 4 (241-300 segundos): valores de UCL e LCL recalculados, conforme condição verificada na janela 3. Taxa de amostragem de 60% com total de 36 amostras selecionadas para análise. Não foram verificados pontos entre os limites de σ_2 e σ_3 , condição que dispara o cálculo de um novo *baseline*.

- f) Janela 5 (301-360 segundos): *baseline* recalculado, conforme condição ocorrida na janela 4. Quantidade de amostras selecionadas para análise é de 36 e taxa de amostragem de 60%. Verificados cinco pontos acima de *sigma3* e dois pontos entre os limites de *sigma2* e 3, condição que dispara novo cálculo de *baseline*.
- g) Janela 6 (361-420 segundos): os valores de UCL e LCL foram recalculados, conforme esperado quanto a quantidade máxima de pontos excede o limite estabelecido no algoritmo. Taxa de amostragem nesta janela é de 60%, com total de 36 amostras analisadas. Nesta janela, não houve pontos entre os limites de *sigma2* e 3, o que dispara, novamente, o cálculo de novo *baseline*.
- h) Janela 7 (421-480 segundos): os valores de UCL e LCL são novos, devido recálculo do *baseline* disparado na janela 6. A quantidade de amostras selecionadas para análise é 36, enquanto a taxa de amostragem é de 60%. Verificados três pontos entre os limites de *sigma2* e 3, condição que mantém o *baseline* atual.
- i) Janela 8 (481-540 segundos): os valores de UCL e LCL são os mesmos calculados na janela 6. Taxa de amostragem em 60% com quantidade de 36 amostras para análise. Verificados seis pontos acima do limite de *sigma3* e três pontos acima do limite de *sigma2*, condição esta, que dispara o recálculo do *baseline*.
- j) Janela 9 (541-600 segundos): os valores de UCL e LCL foram recalculados na janela 8, devido à condição de novo comportamento. Quantidade de amostras selecionadas para análise nesta janela é de 36, com 60% de taxa de amostragem. Verificado um ponto acima do limite de *sigma3* e dois pontos entre os limites de *sigma2* e 3. Como o limite máximo de pontos acima do *sigma3* é de 0.1%, um novo *baseline* foi calculado.
- k) Janela 10 (601-660 segundos): os valores de LCL e UCL foram recalculados na janela 9 devido ao fato da quantidade máxima de pontos ter sido excedida. Taxa de amostragem de 60% com 36 amostras selecionadas para análise. Quantidade de pontos verificado entre limites de *sigma2* e 3 foi um ponto, o que mantém *baseline* para a próxima janela.
- l) Janela 11 (661-720 segundos): os valores de UCL e LCL foram mantidos os mesmos calculados na janela 9, devido à condição verificada na janela 10. Quantidade de amostras selecionadas para análise é de 36, com taxa de amostragem em 60%. Nesta janela, foram verificados três pontos acima do limite de *sigma3* e um ponto entre os limites de *sigma2* e 3.

Neste cenário foi recalculado por diversas vezes o valor de *baseline* devido às alterações no perfil. Na janela 9, verificou-se condição na qual somente um ponto acima do limite de *sigma3* disparou um novo *baseline*, isto se deve ao fato de que a taxa máxima de pontos acima do limite de *sigma3*, conforme estabelecido pela distribuição normal, é de 0.1% dos pontos.

A aderência do gráfico original é de 98.3% enquanto para o gráfico obtido é de 94%.

6.4. CENÁRIO COM MUDANÇA PERMANENTE DE COMPORTAMENTO

Neste cenário, optou-se por gerar uma carga fora do comportamento normal do *link* em questão, de forma que o valor do DF sofresse uma degradação em um período grande tempo, sugerindo assim, uma mudança de perfil. Segue abaixo, os gráficos obtidos:

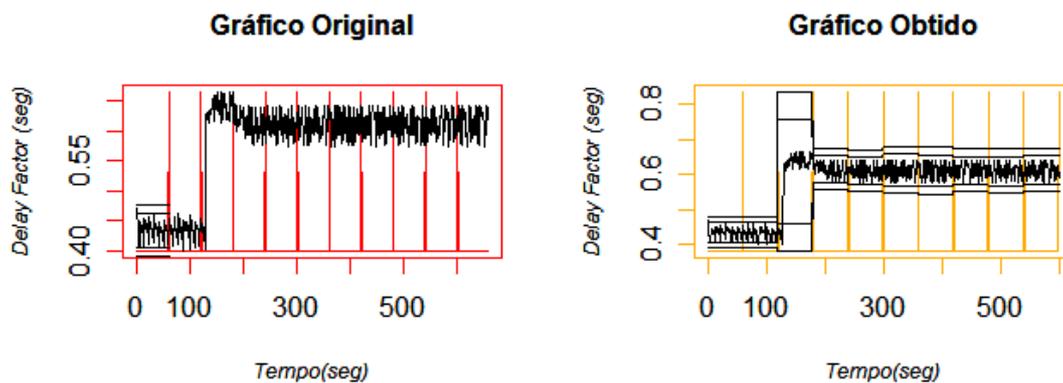


Figura 6.4 Gráfico original e gráfico obtido através do algoritmo de amostragem dinâmica para Cenário com mudança permanente de comportamento

No gráfico original, localizado à esquerda da Figura 6.4, pode ser verificado que o comportamento do tráfego a partir da janela 3 é alterado significativamente. No gráfico obtido, localizado à direita da Figura 6.4, observa-se que as linhas de UCL e LCL foram recalculadas ao longo do tempo. O objetivo deste cenário é visualizar a mudança de comportamento e recalculando os valores dos limites LCL e UCL.

Na tabela 6.4, seguem os dados detalhados por janela de cálculo:

Janela	Índice Inicial	Índice Final	UCL 3	UCL 2	LCL 2	LCL 3	Taxa de Amostragem	Qtde Amostras	Qtde Amostras acima sigma3	Qtde Amostras entre sigma 2 e 3	Qtde Amostras entre média e sigma2
0	0	60	0.4694923	0.4578048	0.4110549	0.3993674	Calculando Baseline				
1	61	120	0.4694923	0.4578048	0.4110549	0.3993674	60%	36	0	2	34
2	121	180	0.4694923	0.4578048	0.4110549	0.3993674	60%	36	0	2	34
3	181	240	0.4694923	0.4578048	0.4110549	0.3993674	60%	36	6	1	29
3	181	240	0.9764281	0.8168006	0.1782904	0.01866291	Calculando Novo Baseline				
4	241	300	0.9764281	0.8168006	0.1782904	0.01866291	60%	36	0	0	36
4	241	300	0.4686973	0.4570204	0.4103129	0.398636	Calculando Novo Baseline				
5	301	360	0.4686973	0.4570204	0.4103129	0.398636	60%	36	5	2	29
5	301	360	0.976218	0.816659	0.1784225	0.01886343	Calculando Novo Baseline				

6	361	420	0.976218	0.816659	0.1784225	0.01886343	60%	36	0	0	36
6	361	420	0.4693746	0.4575807	0.4104049	0.398611	Calculando Novo Baseline				
7	421	480	0.4693746	0.4575807	0.4104049	0.398611	60%	36	0	3	33
8	481	540	0.4693746	0.4575807	0.4104049	0.398611	60%	36	6	3	27
8	481	540	0.9764125	0.8168009	0.1783544	0.01874273	Calculando Novo Baseline				
9	541	600	0.9764125	0.8168009	0.1783544	0.01874273	60%	36	1	2	33
9	541	600	0.857636	0.7287692	0.2133018	0.0844349	Calculando Novo Baseline				
10	601	660	0.857636	0.7287692	0.2133018	0.0844349	60%	36	0	1	35
11	661	720	0.857636	0.7287692	0.2133018	0.0844349	60%	36	3	1	32

Tabela 6.4 Detalhamento dos resultados obtidos para o Cenário de mudança permanente de comportamento

Conforme pode ser visualizado através da tabela 6.4, o algoritmo de amostragem dinâmica teve os seguintes resultados:

- a) Janela 0 (0-60 segundos): valores de UCL e LCL para σ_2 e 3 são calculados e armazenados como *baseline*;
- b) Janela 1 (61-120 segundos): valores de UCL e LCL mantidos, conforme cálculo de *baseline*, efetuado na janela 0. A taxa de amostragem inicial é de 60%, com 36 amostras selecionadas para análise. Nesta janela foram verificados dois pontos entre os limites de σ_2 e 3.
- c) Janela 2 (121-180 segundos): valores de UCL e LCL permanecem os mesmos do *baseline*. Foram selecionadas para análise 36 amostras com taxa de amostragem de 60%. Nesta janela foram verificados dois pontos entre os limites de σ_2 e 3, condição que mantém o valor de *baseline* para a próxima janela.
- d) Janela 3 (181-240 segundos): valor de *baseline* mantido, conforme condição verificada na janela 2. Taxa de amostragem de 60%, com 36 amostras selecionadas para análise. Verificou-se a quantidade de seis pontos acima do limite de σ_3 e um ponto entre os limites de σ_2 e 3, o que dispara o cálculo de novo *baseline*.
- e) Janela 4 (241-300 segundos): valores de UCL e LCL recalculados, conforme condição verificada na janela 3. A quantidade de amostras selecionadas para análise é 36, enquanto a taxa de amostragem é de 60%. Não foram verificados pontos entre os limites de σ_2 e 3, o que dispara o cálculo de um novo *baseline*.
- f) Janela 5 (301-360 segundos): novos valores de UCL e LCL foram calculados, devido à condição verificada na janela 4. Taxa de amostragem de 60% com 36 amostras selecionadas para análise. Foram verificados cinco pontos acima do limite de σ_3 e dois pontos entre os limites de σ_2 e 3, o que novamente, dispara o cálculo do *baseline*.
- g) Janela 6 (361-420 segundos): conforme condição verificada na janela 5, novos valores de UCL e LCL foram calculados para ela. A quantidade de amostras selecionadas para análise é 36, enquanto a taxa de amostragem é de 60%. Não

foram observados pontos entre os limites de σ_2 e 3, disparando um novo *baseline*.

- h) Janela 7 (421-480 segundos): UCL e LCL recalculados, devido a condição observada na janela 6. Taxa de amostragem mantida em 60%. Nesta janela, foram observados três pontos entre os limites de σ_2 e 3, como tal condição está dentro dos limites estabelecidos, o valor de *baseline* será mantido para a próxima janela.
- i) Janela 8 (481-540 segundos): UCL e LCL mantidos conforme cálculos realizados na janela 6. Quantidade de amostras selecionadas para análise é de 36. Foram observados seis pontos acima do limite de σ_3 e entre os limites de σ_2 e 3, condição que dispara cálculo de novo *baseline*.
- j) Janela 9 (541-600 segundos): valores de UCL e LCL recalculados. Taxa de amostragem igual a 60%, totalizando 36 amostras. Nesta janela, foi observado um ponto acima do limite de σ_3 e dois acima do limite de σ_2 e 3, condição que dispara um novo *baseline*, pois os limites mínimos de pontos acima do σ_3 não foram observados.
- k) Janela 10 (601-660 segundos): UCL e LCL recalculados, devido a condição observada na janela 9. Quantidade de amostras selecionadas para análise é de 36. Foi observado somente um ponto entre os limites de σ_2 e 3, condição que mantém o *baseline* calculado na janela 9.
- l) Janela 11 (661-720 segundos): UCL e LCL mantidos, conforme cálculo de *baseline* na janela 9. Taxa de amostragem em 60%, com 36 amostras selecionadas para análise. Nesta janela verificou-se três pontos acima do limite de σ_3 e um ponto entre os limites de σ_2 e 3.

Neste cenário, novos valores de *baseline* foram calculados várias vezes, em alguns casos devido ao fato da própria mudança de comportamento e em outros, onde não houve quantidade mínima de amostras entre os limites de σ_2 e σ_3 e também a quantidade máxima de pontos acima do σ_3 , foi excedida.

A aderência para o gráfico original é de 90% enquanto para o gráfico obtido é de 95%.

7. CONCLUSÃO

O estudo de técnicas de amostragem de dados é bastante relevante, pois oferece uma solução para minimizar a quantidade de dados analisados.

O cálculo do valor do DF foi validado através da observação dos valores desta métrica durante a criação dos cenários de teste. Quando foram criadas condições onde o valor de métrica deveria sofrer alteração, os valores calculados de DF refletiram tais alterações.

Para fins de teste deste algoritmo foram definidos quatro cenários distintos: normal, transitório, recorrente e permanente. A análise dos resultados nos quatro cenários distintos demonstrou que em cenários de comportamento normal o algoritmo utilizou a menor taxa de amostragem possível (30%). Já para os cenários transitório, recorrente e permanente, devido à constante mudança do comportamento da métrica, foi utilizada a amostragem de 60%.

Verificou-se a mudança de perfil da métrica nos cenários onde, em vários momentos, a taxa de amostragem foi de 60%, pois, na definição do algoritmo estipulou-se que, somente após duas janelas consecutivas, com mesmo valor de *baseline*, é diminuída a taxa de amostragem. Desta forma, o algoritmo mostrou-se eficiente considerando que todas as mudanças de comportamento foram detectadas e novos valores de *baseline* foram calculados.

Como proposta para futura pesquisa, recomenda-se a realização de testes com outros limites de sigma, pois os limites 2 e 3 foram estabelecidos inicialmente devido ao utilizado na validação de processos da indústria. Também a realização de testes dos cenários com janelas de tempo com tamanhos inferiores ou superiores ao estabelecido neste trabalho (60 segundos).

8. REFERÊNCIAS BIBLIOGRÁFICAS

AGILENT Technologies, White Paper; **IPTV QoE: Understanding and interpreting MDI values.** France, September 29, 2008.

ALMES, G; KALIDINDI, S. & ZEKAUSKAS, M. **RFC 2679 - A One-way Delay Metric for IPPM.** Request for comments, 1999.

BARBETTA, Pedro Alberto. **Estatística para Cursos de Engenharia e Informática.** São Paulo: Atlas, 2004.

BRADEN, R.; CLARK, D.; SHENKER, S. **RFC 1633 - Integrated Services in the Internet Architecture: an Overview.** Request for comments, 2004.

CISCO Systems. **Understanding aing Delay in Packet Voice Networks.** Disponível em http://www.cisco.com/en/US/tech/tk652/tk698/technologies_white_paper09186a00800a8993.shtml. Acesso em 02/02/2006.

CLAISE, B. (Ed.). **Packet Sampling (PSAMP) Protocol Specifications.** Currently Internet (work in progress) Draft draft-ietf-psamp-protocol-07.txt., 2006.

DIETZ *et al.*, T . **Information Model for Packet Sampling Exports.** Currently Internet Draft, draft-ietf-psamp-info-06, 2007.

DUFFIELD, Nick (Ed.). **A Framework for Packet Selection and Reporting.** Currently Internet Draft (work in progress) draft-ietf-psamp-framework-11, 2007.

ESTAN, C. & VARGHESE, G. **New Directions in Traffic Measurement and Accounting.** *In:* Internet Measurement Workshop. San Francisco: ACM SIGCOMM, 2001.

GUAN, Cheng; JIAN, Gong; WEI, Ding. **A traffic sampling model for measurement using packet identification.** Department of Computer Science & Engineering. Southeast Univ. Nanjing. China: November 7, 2002.

HANEMANN *et al.*, Andreas. **A Study on Network Performance Metrics and their Composition:** (1) German Research Network, (2) Greek Research & Technology Network S.A, (3) DANTE, (4) Dep. of Computer and Information Sciences, University of Delaware. USA: March, 2006.

HEDAYAT, K. (Ed.). **Two-way Active Measurement Protocol (TWAMP).** Currently Internet Draft (work in progress) draft- draft-ietf-ippm-twamp-05, 2007.

INEOQUEST Technologies, Inc. **Media Delivery Indes – Application Note 5.** USA: May 20, 2005a.

INEOQUEST Technologies, Inc. **MDI/QoE para IPTV e VOIP.** USA: July 2, 2005b.

KALIDINDI, S. & ZEKAUSKAS, M. **Surveyor: An Infrastructure for Internet Performance Measurements.** Proceedings of INET'99. San Jose, CA, USA: June 22-25, 1999a.

KALIDINDI, S. & ZEKAUSKAS, M. **RFC 2681 - A Round-trip Delay Metric for IPPM**. USA, 1999b.

KALIDINDI *et al.*, Sunil. **Surveyor**: An Infrastructure for Internet Performance Measurements. INET'99, San Jose, 1999.

MPEG, **The reference website for MPEG**. Disponível em <http://www.mpeg.org/>. Acesso em 10/12/2008.

MONTGOMERY, D. C. **Introduction to Statistical Quality Control**. New York: Wiley, 2005.

NETWORK Metric Report. **DJ1.2.306 - GN2 JRA1 deliverable**. February 2006.

NIST – National Institute of Standards and Technology. **Process or Product Monitoring and Control**. Disponível em <http://www.itl.nist.gov/div898/handbook/pmc/pmc.htm>. Acesso em 10/01/2009.

PAXON *et al.*, V. **RFC 2330 - Framework for IP Performance Metrics**. Request for comments, 2008.

QUITTEK, J. **RFC 3917 - Requirements for IP Flow Information Export**. Request for comments, 2007.

QUITTEK *et al.*, J. **RFC 5102 - Information Model for IP Flow Information Report**. Request for comments, 2008.

SHALUNOV *et al.*, S. **RFC 4656 - One-way Active Measurement Protocol (OWAMP)**. Draft-ietf-ippm-owdp-11.txt. Request for comments, 2004.

SNOEREN *et al.*, A. C. **Hash-Based IP traceback**. *In*: Proceedings of the ACM SIGCOMM 2001. Conference (SIGCOMM-01), volume 31:4 of Computer Communication Review, August 2001.

VETTER, Fausto. **Métricas para Redes**. *In*: Workshop Medições 2. Belém: UFSC, 2007.

WELCH, J. & CLARCK, J. **RFC 4445 - A Proposed Media Delivery Index (MDI)**. Request for comments, 2006.

ZSEBY, Tanja. **Deployment of Sampling Methods for SLA Validation with Non-Intrusive Measurements**. Proceedings of Passive and Active Measurement Workshop (PAM 2002), Fort Collins, CO, USA, March 25-26, 2002.

ZSEBY, Tanja. **Comparison of Sampling Methods for Non-Intrusive SLA Validation**. 2nd Workshop on End-to-End Monitoring Techniques and Services (E2EMON), October 3, 2004

ZSEBY, T.; ZANDER, S. & CARLE, G. **Evaluation of Building Blocks for Passive One-way-delay Measurements**. Proceedings of Passive and Active Measurement Workshop (PAM 2001). Amsterdam: The Netherlands, April 23-24, 2001.

ZSEBY *et al.*, Tanja. **IPFIX Applicability**. IETF Internet Draft (work in progress), draft-ietf-ipfix-as-01.txt, 2004.

ZSEBY *et al.*, Tanja. **Sampling and Filtering Techniques for IP Packet Selection**. Currently Internet Draft (work in progress), draft-ietf-psamp-sample-tech-10.txt., 2007.

9. ANEXO A

ALGORITMO PARA CÁLCULO DO VALOR DO *DELAY FACTOR*

```

Calc_MDI <- function(file_capture,MR) ({
initial_data <- read.table(file_capture, header=TRUE)
attach(initial_data)

# VARIÁVEIS GLOBAIS
i <- 1
interval <- 1
interval_len <- 1
initial_time <- 0
final_time <- initial_time + interval
sample_ini <- 1
sample_len <- Length[i]
sample_last <- length(No.)
Sj <- 0
VBmax <- 0
VBmin <- 0
DF <- numeric(i)
DFinterval <- numeric(i)
VBpre <- numeric(i)
VBpos <- numeric(i)
VBinterval <- numeric(i)
VBpre[i] <- 0
VBpos[i] <- VBpre[i] + sample_len
i <- i+1
# CALCULA VALORES DE VBPre e VBPos
while (i < sample_last) {
if (Time[i+1] <= final_time){
VBpre[i] <- (sample_len - MR) * (Time[i] - (interval - 1))
VBpos[i] <- VBpre[i] + Length[i]
sample_len <- sample_len + Length[i]
}
else {
# GUARDA VALORES DE VBMax e VBMin
VBpre[i] <- (sample_len - MR) * (Time[i] - (interval - 1))
VBpos[i] <- VBpre[i] + Length[i]
VBinterval <- append(VBpre[(sample_ini:i)],VBpos[(sample_ini:i)],after=length(VBpre[(sample_ini:i)]))
VBmax <- max(VBinterval)
VBmin <- min(VBinterval)
DF[interval] <- ((VBmax-VBmin)/(MR))
DFinterval[interval] <- interval
interval <- interval+1
#print(sample_ini)
#print(i)
#print(DF[interval-1])
sample_ini <- i+1
sample_len <- Length[i]
final_time <- final_time+interval_len
while (Time[i+1] > final_time) {
final_time <- final_time+interval_len
DF[interval] <- ((VBmax-VBmin)/(MR))
DFinterval[interval] <- interval
interval <- interval+1
}
}
}
i <- i+1

```

```
}  
print(VBpre)  
print(VBpos)  
print(DFinterval)  
print(DF)  
plot (DFinterval, DF, type = "l", cex.main = 1, font.main= 2, col.main= "black", cex.lab = 0.75, font.lab = 3,  
col.sub = "red", main = "Cálculo de Delay Factor(DF)", xlab="Tempo(seg)", ylab="Delay Factor (seg)")  
})  
Calc_MDI("Fsrj1_001ncz_6037 frames.txt",1394)  
)
```

10. ANEXO B

ALGORITMO DE AMOSTRAGEM DINÂMICA

```

rm(list=ls(all=TRUE))

tabelaArquivo <- read.table(file="permanente.txt" )
#tabelaArquivo <- read.table(file="testeRecorrente.txt" )
#tabelaArquivo <- read.table(file="transitorio.txt" )
#tabelaArquivo <- read.table(file="normal.txt" )

#HABILITA VARIOS GRAFICOS EM UMA JANELA APENAS
par(mfrow=c(1,2))
par(fg="#000000")

#VARIAVES GLOBAIS
valoresMDI <- tabelaArquivo[,]
janela <- 60
sigma2 <- 2
sigma3 <- 3
janela <- 60
porcentagemDeAmostragemAlta <- 60
porcentagemDeAmostragemBaixa <- 30
quantiaDeQuadrosNecessarioParaAmostragemBaixa <- 2
porcentagemDeAmostragemUtilizada <- porcentagemDeAmostragemAlta
graficoTotalObtido <- c(1)
graficoJanelaObtido <- c(1)
graficoUCL3 <- c(1)
graficoUCL2 <- c(1)
graficoLCL2 <- c(1)
graficoLCL3 <- c(1)
limiteMaxPercentualFora <- 0.1
limiteMinPercentualMeio <- 2.1
limiteMaxPercentualDentro <- 47.7
rendimentoAbsolutoOriginal <- 0
rendimentoAbsolutoObtido <- 0
rendimentosObtidos <- c(1)
rendimentoOriginalPorQuadro <- c(1)

ultimoAmostraGraficoObtidoQuadroAnterior <- valoresMDI[1]
contadorQuadrosDentroBaseline <- 0
repetirProcessamentoJanelaAtual <- "nao"
tentativasEndireitarSinalBaixo <- 0
maximoTentativasEndireitarSinalBaixo <- 1

###

#cALCULANDO UCLS PARA BASELINE INICIAL
mediaMDI <- mean(valoresMDI[1:janela])
desvioPadrao <- sd(valoresMDI[1:janela])

UCL2 <- mediaMDI + sigma2 * desvioPadrao
LCL2 <- mediaMDI - sigma2 * desvioPadrao
UCL3 <- mediaMDI + sigma3 * desvioPadrao
LCL3 <- mediaMDI - sigma3 * desvioPadrao
###

#CALCULANDO ADERÊNCIA PARA JANELA DO BASELINE
qtdeAmostrasFora <- 0
qtdeAmostrasDentro <- 0

```

```

for( ind in 1 : janela ){
  if( valoresMDI[ind] > UCL3 || valoresMDI[ind] < LCL3 ){
    qtdeAmostrasFora <- qtdeAmostrasFora + 1
  }else{
    qtdeAmostrasDentro <- qtdeAmostrasDentro + 1
  }
}
rendimentoJanelaBaseline = (qtdeAmostrasDentro * 100 / janela )
rendimentosObtidos[length(rendimentosObtidos) + 1 ] <- rendimentoJanelaBaseline
###

#CALCULANDO ADERÊNCIA PARA GRAFICO INTEIRO
qtdeAmostrasFora <- 0
qtdeAmostrasDentro <- 0
for( ind in 1 : length(valoresMDI) ){
  if( valoresMDI[ind] > UCL3 || valoresMDI[ind] < LCL3 ){
    qtdeAmostrasFora <- qtdeAmostrasFora + 1
  }else{
    qtdeAmostrasDentro <- qtdeAmostrasDentro + 1
  }
}
rendimentoAbsolutoOriginal = qtdeAmostrasDentro * 100 / (length( valoresMDI ) )
###

#CALCULA O NÚMERO DE QUADRANTES NO GRAFICO ORIGINAL COMPLETO
numeroDeQuadrantes <- length(valoresMDI) / janela
print("Numero de quadrantes")
print( numeroDeQuadrantes )
###

#CALCULA ADERÊNCIA INDIVIDUAL PARA CADA JANELA COM BASE NOS UCLs e LCLs da
primeira janela
for( ind in 0 : numeroDeQuadrantes ){
  indiceInicialJanela <- ind * janela
  indiceFinalJanela <- indiceInicialJanela + janela
  qtFora <- 0
  qtDentro <- 0

  if( ind == 0 ){
    indiceInicialJanela <- 1
    indiceFinalJanela <- janela
  }
  for( id in indiceInicialJanela : indiceFinalJanela ){
    if(valoresMDI[id] > UCL3 || valoresMDI[id] < LCL3 ){
      qtFora <- qtFora + 1
    }else{
      qtDentro <- qtDentro + 1
    }
  }
  rendimentoOriginalPorQuadro[length(rendimentoOriginalPorQuadro) + 1] <- (qtDentro * 100 / janela)
}

print( "#####" )
print( "Valores para o baseline atual, quadrante ZERO" )
print( "Amostras | Dentro | Fora | rendimentoJanelaBaseline" )

```

```

print( janela )
print( qtdeAmostrasDentro )
print( qtdeAmostrasFora )
print( rendimentoJanelaBaseline )
print( "#####" )

linhaDivisoriaJanelas <-c(1)
for(ind in 1 : length(valoresMDI) ){
  linhaDivisoriaJanelas[ind] <- min(valoresMDI)
}
for(ind in 1 : numeroDeQuadrantes){
  linhaDivisoriaJanelas[ ind * janela ] <- max(valoresMDI)
}
t <- 1:length(linhaDivisoriaJanelas)
par(fg="#FF0000")
plot(type="l", ylab="", xlab="", t, linhaDivisoriaJanelas )
par(fg="#000000")

#APRESENTA O GRAFICO ORIGINAL
ttempo <- c(1 : length(valoresMDI))
lines(xlab="", ylab="",ttempo, valoresMDI)
title("Gráfico Original",xlab="Tempo(seg)", ylab="Delay Factor (seg)", cex.main= 1, font.main= 2, col.main=
"black", cex.lab = 0.75, font.lab = 3, col.sub = "red")
#Sys.sleep(2)
##

#for( indice in 1 : length(valoresMDI) ){
#  graficoUCL3[indice] <- UCL3
#  graficoUCL2[indice] <- UCL2
#  graficoLCL2[indice] <- LCL2
#  graficoLCL3[indice] <- LCL3
#}

graficoTotalObtido      <- c(valoresMDI[1:janela])
for( indice in 1 : janela ){
  graficoUCL3[indice] <- UCL3
  graficoUCL2[indice] <- UCL2
  graficoLCL2[indice] <- LCL2
  graficoLCL3[indice] <- LCL3
}
ttt <- 1:janela
lines(ttt, graficoUCL3 )
lines(ttt, graficoUCL2 )
lines(ttt, graficoLCL2 )
lines(ttt, graficoLCL3 )

# PERCORRE TODOS OS QUADRANTES
offSetFor <- 0
for( indiceJanelaAtual in 1 : (numeroDeQuadrantes * 2 - 1)){
  if( repetirProcessamentoJanelaAtual == "sim" ){
    offSetFor <- offSetFor + 1
  }
  indiceJanelaAtual <- indiceJanelaAtual - offSetFor

  print( "=====QUADRANTE=====")
}

```

```

print( indiceJanelaAtual )

#CONTEM O INDICE DE AMOSTRA INICIAL E FINAL PARA A JANELA ATUAL
indiceAmostraInicialJanela <- indiceJanelaAtual * janela + 1
indiceAmostraFinalJanela = indiceAmostraInicialJanela + janela - 1
repetirProcessamentoJanelaAtual <- "nao"
print("IndiceInicialJanela IndiceFinalJanela")
print(indiceAmostraInicialJanela)
print(indiceAmostraFinalJanela )
print( "UCL3 UCL2 LCL2 LCL3")
print( UCL3 )
print( UCL2 )
print( LCL2 )
print( LCL3 )

#APRESENTA O GRAFICO ORIGINAL DO QUADRANTE ATUAL
graficoOriginalQuadranteAtual <- c(1)
indice <- 0
for( ind in indiceAmostraInicialJanela : indiceAmostraFinalJanela ){
    graficoOriginalQuadranteAtual[indice] <- valoresMDI[ind]
    indice <- indice + 1
}
#ttempo <- c(1 : length(graficoOriginalQuadranteAtual))
#plot(type="l", xlab="Original", ylab=multiplicadorJanela, ttempo, graficoQuadrante)
#Sys.sleep(2)
##/##

#POPULA valoresAnalisados COM AS AMOSTRAS A SEREM COLETADAS E
# idAmostras COM OS INDICES DE AMOSTRAS QUE SERÃO COLETADOS

print("====Porcentagem Utilizada" )
print( porcentagemDeAmostragemUtilizada )

#REPRESENTA A QUANTIDADE DE AMOSTRAS QUE SERÃO COLHIDAS PARA ESTA
JANELA
qtdeAmostrasColhidas <- ( length( valoresMDI[indiceAmostraInicialJanela :
indiceAmostraFinalJanela] ) * porcentagemDeAmostragemUtilizada ) / 100

#REPRESENTA O INTERVALO QUE ESTAS AMOSTRAS SERÃO COLHIDAS DO GRAFICO
ORIGINAL
frequenciaAmostragem <- length( valoresMDI[indiceAmostraInicialJanela : indiceAmostraFinalJanela
] ) / qtdeAmostrasColhidas

idAmostras <- c(1)

#PREENCHE idAmostras COM OS INDICES DO QUADRO ATUAL QUE CONTEM AS
AMOSTRAS QUE SERÃO COLHIDAS
#print("Ids de amostras que serao colhidas")
for( x in 1 : qtdeAmostrasColhidas ){
    idAmostras[x] <- round( x * frequenciaAmostragem + indiceAmostraInicialJanela )
}
#print( idAmostras )

print("ValoresTodosdaJanela")
print(valoresMDI[indiceAmostraInicialJanela:indiceAmostraFinalJanela])
#PREENCHE valoresAnalisados COM AS AMOSTRAS QUE SERÃO ANALISADAS
valoresAnalisados <- c(1)

```

```

print("Valores analisados")
for( ind in 1 : length( idAmostras ) ){
    valoresAnalisados[ind] <- valoresMDI[idAmostras[ind]]
}
print(valoresAnalisados)
###

#OBTEM O NUMERO DE AMOSTRAS DENTRO,FORA E ENTRE OS VALORES DE SIGMA
ESTABELECIDOS
porcentagemAmostrasFora <- 0
porcentagemAmostrasMeio <- 0
porcentagemAmostrasDentro <- 0
qtdeAmostrasFora <- 0
qtdeAmostrasMeio <- 0
qtdeAmostrasDentro <- 0
amostrasMeio <- 0
amostrasFora <- 0
amostrasDentro <- 0
arrayResultado <- c(1)
graficoObtido <- c(1)
for( ind in 1 : length( valoresAnalisados - 1 ) ){

    if( valoresAnalisados[ind] > UCL3 || valoresAnalisados[ind] < LCL3 ){
        qtdeAmostrasFora <- qtdeAmostrasFora + 1
        amostrasFora[ length(amostrasFora) + 1 ] <- valoresAnalisados[ind]
    }else{
        if( (valoresAnalisados[ind] < UCL3 && valoresAnalisados[ind] > UCL2) ||
            (valoresAnalisados[ind] > LCL3 && valoresAnalisados[ind] < LCL2) ){
            qtdeAmostrasMeio <- qtdeAmostrasMeio+ 1
            amostrasMeio[ length(amostrasMeio) + 1 ] <- valoresAnalisados[ind]
        }else{
            qtdeAmostrasDentro <- qtdeAmostrasDentro + 1
            amostrasDentro[ length(amostrasDentro) + 1 ] <- valoresAnalisados[ind]
        }
    }
}
porcentagemAmostrasFora <- ( qtdeAmostrasFora * 100 ) / qtdeAmostrasColhidas
porcentagemAmostrasMeio <- ( qtdeAmostrasMeio * 100 ) / qtdeAmostrasColhidas
porcentagemAmostrasDentro <- ( qtdeAmostrasDentro * 100 ) / qtdeAmostrasColhidas
###

#APRESENTA ESTATISTICAS PARA A JANELA ATUAL
print( "Quantidade de amostras colhidas" )
print( qtdeAmostrasColhidas )

print( "Qtde_Fora %_Fora" )
print( qtdeAmostrasFora )
print( porcentagemAmostrasFora )
print( "AmostrasFora" )
print( amostrasFora )

print( "Qtde_Meio %_Meio" )
print( qtdeAmostrasMeio )
print( porcentagemAmostrasMeio )
print( "AmostrasMeio" )
print( amostrasMeio )

print( "Qtde_Dentro %_Dentro" )

```

```

print( qtdeAmostrasDentro )
print( porcentagemAmostrasDentro )
print( "AmostrasDentro" )
print( amostrasDentro )

print( "Limite_MAX_Fora Limite_MIN__Meio Limite_Minimo_Dentro" )
print( limiteMaxPercentualFora )
print( limiteMinPercentualMeio )
print( limiteMaxPercentualDentro )
##

#NECESSÁRIO PARA OBTER O GRAFICO DAS AMOSTRAS NESTA JANELA
indiceParaGraficoObtido <- 1
#PERCORRE A JANELA ORIGINAL
for( ind in indiceAmostraInicialJanela : indiceAmostraFinalJanela ){

    # VERIFICA SE O INDICE É O PROCURADO PARA AMOSTRAGEM
    indiceAmostradoProcurado <- "nao"
    for( indAm in 1 : length(idAmostras)){
        if( idAmostras[indAm] == ind && valoresMDI[idAmostras[indAm]] < UCL3 &&
valoresMDI[idAmostras[indAm]] > LCL3 ){
            #if( idAmostras[indAm] == ind ){
                indiceAmostradoProcurado <- "sim"
                valorAmostraAtual <- valoresMDI[idAmostras[indAm]]
            }
        }
    }
    ##

    if( indiceAmostradoProcurado == "sim" ){
        graficoJanelaObtido[ indiceParaGraficoObtido ] <- valorAmostraAtual
    }else{
        #CASO O INDICE ATUAL NÃO SEJA UMA AMOSTRA PROCURADA,
        #SE NAO FOR O PRIMEIRO INDICE DESTE GRAFICO, DEVE REPETIR O
VALOR DA AMOSTRAGEM ANTERIOR
        if( indiceParaGraficoObtido > 1 ){
            indiceAnterior <- indiceParaGraficoObtido - 1
            graficoJanelaObtido[ indiceParaGraficoObtido ] <- graficoJanelaObtido[
indiceAnterior ]
        }else{
            #SE ESTE FOR O PRIMEIRO INDICE, E NAO FOR UMA AMOSTRA
PROCURADA, DEVE OBTER O
            # VALOR ORIGINAL DO INDICE INICIAL CASO ESTE SEJA ENTRE
LIMITES, OU O piso/teto do baseline
            if( valoresMDI[indiceAmostraInicialJanela] > LCL3 &&
valoresMDI[indiceAmostraInicialJanela] < UCL3 ){
                graficoJanelaObtido[ indiceParaGraficoObtido ] <-
valoresMDI[indiceAmostraInicialJanela]
            }else{
                if( valoresMDI[ind] > UCL3 ){
                    graficoJanelaObtido[ indiceParaGraficoObtido ] <- UCL3
                }

                if( valoresMDI[ind] < LCL3 ){
                    graficoJanelaObtido[ indiceParaGraficoObtido ] <- LCL3
                }
            }
        }
    }
}

```

```

    }
  }
  indiceParaGraficoObtido <- indiceParaGraficoObtido + 1
}
ultimoAmostraGraficoObtidoQuadroAnterior <- graficoJanelaObtido[ length(graficoJanelaObtido) - 1 ]

#print("GraficoJanelaObtido")
#print( graficoJanelaObtido)
###

janelaValidada <- "nao"

contadorQuadrosDentroBaseline <- contadorQuadrosDentroBaseline + 1

if( janelaValidada == "nao" ){
  #VERIFICA SE PERFIL DAS AMOSTRAS OBTIDAS É MAIOR QUE limiteMaxPercentualFora
  if( porcentagemAmostrasFora > limiteMaxPercentualFora ){
    #DEFINIR NOVO BASELINE
    contadorQuadrosDentroBaseline <- 0
    media <- mean(valoresMDI[indiceAmostraInicialJanela:indiceAmostraFinalJanela])
    desvioPadrao <-
sd(valoresMDI[indiceAmostraInicialJanela:indiceAmostraFinalJanela])
    UCL2 <- media + sigma2 * desvioPadrao
    LCL2 <- media - sigma2 * desvioPadrao
    UCL3 <- media + sigma3 * desvioPadrao
    LCL3 <- media - sigma3 * desvioPadrao
    ###
    print("RECALCULANDO BASELINE-Perfil acima de limites sigma3")
    print("Inicial/final media/desvioPadrao UCL3/UCL2 LCL2/LCL3")
    print( indiceAmostraInicialJanela )
    print( indiceAmostraFinalJanela )
    print( media )
    print( desvioPadrao )
    print( UCL3 )
    print( UCL2 )
    print( LCL2 )
    print( LCL3 )
    print("###")
    #GRAFICO OBTIDO DESTA JANELA É A PROPRIA JANELA ORIGINAL
    graficoJanelaObtido <-
valoresMDI[indiceAmostraInicialJanela:indiceAmostraFinalJanela]
    janelaValidada <- "sim"
  }
}
###

if( janelaValidada == "nao" ){
  #VERIFICA SE PERFIL DAS AMOSTRAS OBTIDAS É MENOR QUE limiteMaxPercentualFora
  # E MAIOR QUE LIMITEMINPERCENTUALMEIO
  if( porcentagemAmostrasFora > limiteMaxPercentualFora || porcentagemAmostrasMeio >
limiteMinPercentualMeio){
    #NAO DEVE DEFINIR NOVO BASELINE
    #DEVE INCREMENTAR contadorQuadrosDentroBaseline

```

```

contadorQuadrosDentroBaseline <- contadorQuadrosDentroBaseline + 1

print("PERCENTUAIS DENTRO DOS LIMITES DE SIGMA 2 E 3")

janelaValidada <- "sim"
}
}
###

if( janelaValidada == "nao" ){
  #VERIFICA SE PERFIL DAS AMOSTRAS OBTIDAS É MAIOR QUE limiteMaxPercentualFora
  if( porcentagemAmostrasMeio < limiteMinPercentualMeio &&
    porcentagemAmostrasDentro > limiteMaxPercentualDentro){
    if( tentativasEndireitarSinalBaixo <=
maximoTentativasEndireitarSinalBaixo ){
      tentativasEndireitarSinalBaixo <- tentativasEndireitarSinalBaixo + 1
      #SUBIR TAXA DE AMOSTRAGEM E REPETIR O TRATAMENTO
PARA ESTA JANELA
      print("COMPORTAMENTO MUITO ABAIXO DE REQUISITADO DE
SIGMA 2" )

      print("SUBINDO TAXA DE AMOSTRAGEM")
      print("REPETINDO PROCESSAMENTO PARA ESTA JANELA")
      contadorQuadrosDentroBaseline <- 0
      repetirProcessamentoJanelaAtual = "sim"
      tentativasEndireitarSinalBaixo <- tentativasEndireitarSinalBaixo + 1
      janelaValidada <- "sim"
      indiceJanelaAtual <- indiceJanelaAtual -1
    }else{
      print ("JA TENTADO AJUSTAR COMPORTAMENTO DE SINAL
BAIXO")

      print ("RECALCULANDO BASELINE")
      contadorQuadrosDentroBaseline <- 0
      media <-
mean(valoresMDI[indiceAmostraInicialJanela:indiceAmostraFinalJanela])
      desvioPadrao <-
sd(valoresMDI[indiceAmostraInicialJanela:indiceAmostraFinalJanela])
      UCL2 <- media + sigma2 * desvioPadrao
      LCL2 <- media - sigma2 * desvioPadrao
      UCL3 <- media + sigma3 * desvioPadrao
      LCL3 <- media - sigma3 * desvioPadrao
      ###
      print("Inicial/final media/desvioPadrao UCL3/UCL2 LCL2/LCL3")
      print( indiceAmostraInicialJanela )
      print( indiceAmostraFinalJanela )
      print( media )
      print( desvioPadrao )
      print( UCL3 )
      print( UCL2 )
      print( LCL2 )
      print( LCL3 )
      print("###" )
      #GRAFICO OBTIDO DESTA JANELA É A PROPRIA JANELA
ORIGINAL
      graficoJanelaObtido <-
valoresMDI[indiceAmostraInicialJanela:indiceAmostraFinalJanela]
      janelaValidada <- "sim"

```

```

        tentativasEndireitarSinalBaixo <- 0
    }

}

}

###

if( janelaValidada == "nao" ){
    #VERIFICA SE PERFIL DAS AMOSTRAS OBTIDAS É MENOR QUE
    limiteMaxPercentualDentro
    if( porcentagemAmostrasDentro > limiteMaxPercentualDentro ){
        #DEFINIR NOVO BASELINE
        contadorQuadrosDentroBaseline <- 0
        media <- mean(valoresMDI[indiceAmostraInicialJanela:indiceAmostraFinalJanela])
        desvioPadrao <-
sd(valoresMDI[indiceAmostraInicialJanela:indiceAmostraFinalJanela])
        UCL2 <- media + sigma2 * desvioPadrao
        LCL2 <- media - sigma2 * desvioPadrao
        UCL3 <- media + sigma3 * desvioPadrao
        LCL3 <- media - sigma3 * desvioPadrao
        ###
        print("RECALCULANDO BASELINE-Perfil muito abaixo de sigma2")
        print("Inicial/final media/desvioPadrao UCL3/UCL2 LCL2/LCL3")
        print( indiceAmostraInicialJanela )
        print( indiceAmostraFinalJanela )
        print( media )
        print( desvioPadrao )
        print( UCL3 )
        print( UCL2 )
        print( LCL2 )
        print( LCL3 )
        print("##" )
        #GRAFICO OBTIDO DESTA JANELA É A PROPRIA JANELA ORIGINAL
        graficoJanelaObtido <-
valoresMDI[indiceAmostraInicialJanela:indiceAmostraFinalJanela]
        janelaValidada <- "sim"
    }
}

###

if( contadorQuadrosDentroBaseline == 0 ){
    porcentagemDeAmostragemUtilizada <- porcentagemDeAmostragemAlta
}

if( contadorQuadrosDentroBaseline > quantiaDeQuadrosNecessarioParaAmostragemBaixa ){
    print("Diminuindo taxa de amostragem...")
    porcentagemDeAmostragemUtilizada <- porcentagemDeAmostragemBaixa
}

if( repetirProcessamentoJanelaAtual == "nao" ){
    #ACRESCENTA AO GRAFICO FINAL, O GRAFICO OBTIDO COM ESTA JANELA
    graficoTotalObtido <- append(graficoTotalObtido, graficoJanelaObtido,
after=length(graficoTotalObtido))
}

```

```

#CALCULANDO ADERÊNCIA PARA GRAFICO OBTIDO NESTA JANELA
qtdeAmostrasFora <- 0
qtdeAmostrasDentro <- 0
for( ind in indiceAmostraInicialJanela : indiceAmostraFinalJanela ){
  if( valoresMDI[ind] > UCL3 || valoresMDI[ind] < LCL3 ){
    qtdeAmostrasFora <- qtdeAmostrasFora + 1
  }else{
    qtdeAmostrasDentro <- qtdeAmostrasDentro + 1
  }
}
rendimentosObtidos[length(rendimentosObtidos) + 1 ] <- qtdeAmostrasDentro * 100 / janela
##
print( UCL3)
for( indice in length(graficoUCL3) : (length(graficoUCL3) + janela) ){
  graficoUCL3[indice] <- UCL3
  graficoUCL2[indice] <- UCL2
  graficoLCL2[indice] <- LCL2
  graficoLCL3[indice] <- LCL3
}
}

#tempo <- c(1)
#tempo <- c(1:length(graficoJanelaObtido))
#plot(type="l", tempo, graficoJanelaObtido )

#PULA PARA A PROXIMA JANELA
# if( repetirProcessamentoJanelaAtual == "nao" ){
#   indiceJanelaAtual <- indiceJanelaAtual + 1
# }
# }
soma <- 0
for( ind in 2 : length( rendimentosObtidos ) ){
  soma <- soma + rendimentosObtidos[ind]
}

rendimentoAbsolutoObtido <- soma / ( length(rendimentosObtidos) - 1 )

print("#####")
print("Aderência para o gráfico Inteiro com base no primeiro Baseline")
print(rendimentoAbsolutoOriginal)

print("Aderência para o gráfico Inteiro depois do tratamento")
print(rendimentoAbsolutoObtido)

print("Aderências individuais para cada janela do gráfico original")
print( rendimentoOriginalPorQuadro[2:length(rendimentoOriginalPorQuadro)] )

print("Aderências individuais para cada janela do gráfico resultante")
print( rendimentosObtidos[2:length(rendimentosObtidos)] )
print("#####")

graficoTotalObtido[1] <- valoresMDI[1]
tempo <- c(1)
tempo <- c(1:length(graficoTotalObtido))

```

```

#plot(type="l", ylab="", xlab="", tempo, graficoTotalObtido )
#plot(type="l", ylab="", xlab="", tempo, graficoTotalObtido )
ttt <- 1:length(graficoUCL3)

#lines(ttt,graficoUCL3)
#lines(ttt,graficoUCL2)
#lines(ttt,graficoLCL2)
#lines(ttt,graficoLCL3)
#title("",sub=sigma3,xlab=sigma2)
linhaDivisoriaJanelas <-c(1)
for(ind in 1 : length(graficoTotalObtido) ){
  linhaDivisoriaJanelas[ind] <- min(graficoLCL3)
}
for(ind in 1 : numeroDeQuadrantes){
  linhaDivisoriaJanelas[ ind * janela ] <- max(graficoUCL3)
}
t <- 1:length(linhaDivisoriaJanelas)
par(fg="#FFAA00")
plot(type="l", ylab="", xlab="", tempo, linhaDivisoriaJanelas )
par(fg="#000000")
lines(ttt,graficoUCL3)
lines(ttt,graficoUCL2)
lines(ttt,graficoLCL2)
lines(ttt,graficoLCL3)
title("Gráfico Obtido",xlab="Tempo(seg)", ylab="Delay Factor (seg)", cex.main = 1, font.main= 2, col.main=
"black", cex.lab = 0.75, font.lab = 3, col.sub = "red")
lines(t,graficoTotalObtido)

```