

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**MODELAGEM, GERAÇÃO E ANÁLISE DE TRÁFEGOS EM
REDES MULTISERVIÇOS**

**BRUNO GIANCRISTOFORO QUEIROZ
BRENO NERI CARNEIRO**

**ORIENTADOR: PAULO HENRIQUE PORTELA DE
CARVALHO**

**PROJETO FINAL DE GRADUAÇÃO EM ENGENHARIA DE
REDES DE COMUNICAÇÃO**

BRASÍLIA / DF: AGOSTO/2006

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**MODELAGEM, GERAÇÃO E ANÁLISE DE TRÁFEGOS EM
REDES MULTISERVIÇOS.**

**BRUNO GIANCRISTOFORO QUEIROZ
BRENO NERI CARNEIRO**

PROJETO FINAL DE GRADUAÇÃO SUBMETIDO AO DEPARTAMENTO DE
ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE
BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO
GRAU DE ENGENHEIRO.

APROVADA POR:

(DOUTOR) PAULO HENRIQUE PORTELA DE CARVALHO
(Orientador)

PRISCILA A.S.M. BARRETO
(Examinador Interno)

(DOUTOR) RICARDO STARCIARINI PUTTINI
(Examinador Interno)

ANTONIO JOSE RIBEIRO DOS SANTOS
(Examinador Interno)

MARCIO AUGUSTO DE DEUS
(Examinador Interno)

DATA: BRASÍLIA/DF, 23 de agosto de 2006.

FICHA CATALOGRÁFICA

CARNEIRO, BRENO NERI
QUEIROZ, BRUNO GIANCRISTOFORO
MODELAGEM, GERAÇÃO E ANÁLISE DE TRÁFEGOS EM REDES MULTISERVIÇOS. [Distrito Federal].

xi, 89p, 297mm (ENE/FT/UnB, Engenheiro, Engenharia de Redes de Comunicação, 2006).

Projeto Final de Graduação – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

REFERÊNCIA BIBLIOGRÁFICA

- CARNEIRO, BRENO NERI e QUEIROZ, BRUNO GIANCRISTOFORO (2006). MODELAGEM, GERAÇÃO E ANÁLISE DE TRÁFEGOS EM REDES MULTISERVIÇOS (Projeto Final de Graduação), Publicação UnB.LabRedes.PFG.001/2006, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 89p.

CESSÃO DE DIREITOS

-
- NOME DO AUTOR: Breno Neri Carneiro e Bruno Giancristoforo Queiroz
- TÍTULO DO PROJETO FINAL: MODELAGEM, GERAÇÃO E ANÁLISE DE TRÁFEGOS EM REDES MULTISERVIÇOS
- GRAU/ANO: Engenheiro/2006.
-
-

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Projeto Final de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação de Projeto Final pode ser reproduzida sem a autorização por escrito dos autores.

Breno Neri Carneiro
SQN 312 Bloco G Apartamento 608
CEP 70765-070 – Brasília – DF – Brasil

Bruno Giancristoforo Queiroz
SQN 311 Bloco I Apartamento 504
CEP 70757-090 – Brasília – DF – Brasil

Agradecimentos

À nossas famílias, que sempre nos apoiaram e acreditaram nas nossas capacidades.

Ao orientador, Prof. Dr. Paulo Henrique Portela de Carvalho, que sempre demonstrou confiança em nós e por isso sempre exigiu o nosso melhor.

Aos que compartilharam desse projeto e que foram fundamentais para o resultado que atingimos: Renato Sales Bizerra, Priscila A.S.M. Barreto e Márcio Augusto de Deus.

Aos colegas do labcom e do lemon pela amizade e cooperação.

Aos amigos que fizemos durante esses anos de curso e que colaboraram para o que somos hoje.

E, principalmente, a Deus, por tudo que nos concedeu em nossas vidas e especialmente pelo privilégio de conviver com todas essas pessoas.

Resumo

Este trabalho apresenta os novos modelos utilizados para tráfegos em redes multiserviços, mostrando suas características e implicações. Também mostramos métodos para gerarmos tráfegos de acordo com esses modelos e caracterizarmos tráfegos reais com relação às propriedades dos modelos. Buscamos com esse trabalho estabelecer bases para auxiliar no dimensionamento de uma rede de acordo com o tipo de tráfego nela verificado.

Abstract

This work presents new models for network traffics, showing their characteristics and implications. We also show methods for generating traffics accordingly to these models and for characterizing real traffics based on these model's properties. We intend, in this work, to establish bases to help the dimensioning of a network according to its traffic.

Sumário

AGRADECIMENTOS.....	IV
SUMÁRIO	VII
LISTA DE FIGURAS	IX
LISTA DE TABELAS	XI
1. INTRODUÇÃO.....	1
2. TRÁFEGO AUTO-SIMILAR	5
2.1. <i>AUTO-SIMILARIDADE</i>	<i>6</i>
2.2. <i>DEPENDÊNCIA DE LONGA DURAÇÃO</i>	<i>9</i>
2.3. <i>AUTO-SIMILARIDADE EM TEMPO DISCRETO.....</i>	<i>10</i>
2.4. <i>MOVIMENTO BROWNIANO FRACIONÁRIO (FBM) E RUÍDO GAUSSIANO FRACIONÁRIO (FGN)</i>	<i>14</i>
3. TRÁFEGO MULTIFRACTAL	15
3.1. <i>CASCATAS MULTIPLICATIVAS.....</i>	<i>20</i>
4. INFERÊNCIA.....	22
4.1. <i>ESTIMAÇÃO DE MÉDIA E VARIÂNCIA</i>	<i>28</i>
4.2. <i>ESPECIFICAÇÃO DE DISTRIBUIÇÕES TEÓRICAS BASEADA EM DADOS OBSERVADOS.....</i>	<i>29</i>
4.3. <i>ESTIMAÇÃO DO PARÂMETRO DE HURST</i>	<i>33</i>
4.3.1. <i>Estimador Whittle.....</i>	<i>34</i>
4.3.2. <i>Método Variance-Time Plot.....</i>	<i>35</i>
4.3.3. <i>Método de Kettani-Gubner.....</i>	<i>35</i>
4.3.4. <i>Método do Periodograma</i>	<i>36</i>
5. GERAÇÃO DE TRÁFEGO	37
5.1. <i>GERAÇÃO POR INVERSÃO</i>	<i>38</i>
5.2. <i>GERAÇÃO DE PROCESSO FGN.....</i>	<i>38</i>
5.3. <i>WIG – WAVELET-DOMAIN INDEPENDENT GAUSSIAN MODEL</i>	<i>41</i>
5.4. <i>MWM – MULTIFRACTAL WAVELET MODEL</i>	<i>42</i>
6. GTAR.....	45
7. RESULTADOS OBTIDOS	50
7.1. <i>TESTES DE GERAÇÃO</i>	<i>50</i>
7.1.1. <i>Geração de Tráfego Poisson.....</i>	<i>50</i>
7.1.2. <i>Geração de Tráfego Auto-similar</i>	<i>51</i>
7.2. <i>ESTIMAÇÃO DE TRÁFEGOS REAIS.....</i>	<i>51</i>
7.3. <i>SIMULAÇÃO DE FILA</i>	<i>56</i>
8. CONCLUSÃO	60
APÊNDICES	62
A TRÁFEGO POISSON	63
A.1 <i>PROCESSO DE POISSON</i>	<i>63</i>
A.2 <i>PROPRIEDADES DO PROCESSO DE POISSON</i>	<i>67</i>
A.3 <i>PROPRIEDADE “MARKOVIANA” DA DISTRIBUIÇÃO EXPONENCIAL</i>	<i>68</i>
B TRÁFEGO RAJADA.....	70
C TEORIA DE FILAS.....	72
C.1 <i>NOTAÇÃO DE KENDALL</i>	<i>72</i>
C.2 <i>TEOREMA DE LITTLE</i>	<i>73</i>
C.3 <i>MODELAGEM PARA FILA M/M/1</i>	<i>75</i>
C.4 <i>MODELAGEM PARA FILAS COM NÚMERO DE USUÁRIOS EM ESPERA LIMITADO (M/M/1/K).....</i>	<i>77</i>
C.5 <i>TEOREMA DA INDEPENDÊNCIA DE KLEINROCK.....</i>	<i>78</i>
D MULTIRESOLUÇÃO	81

<i>D.1</i>	<i>WAVELETS DE HAAR</i>	84
------------	-------------------------------	----

Lista de Figuras

Figura 1.1 – Diagrama de um sistema de otimização de rede	3
Figura 2.1 – Comparação entre um tráfego poissoniano e um tráfego real (extraída de [6])	6
Figura 3.1 – Construção de uma cascata binomial	20
Figura 3.2 – Função amostra criada utilizando-se o método da cascata binomial.....	21
Figura 4.1 – Distribuição de cauda do atraso com o processo de chegada tendo parâmetro de Hurst igual a 0,5.....	25
Figura 4.2 - Distribuição de cauda do atraso com o processo de chegada tendo parâmetro de Hurst igual a 0,6	25
Figura 4.3 - Distribuição de cauda do atraso com o processo de chegada tendo parâmetro de Hurst igual a 0,7	26
Figura 4.4 - Distribuição de cauda do atraso com o processo de chegada tendo parâmetro de Hurst igual a 0,8	26
Figura 4.5 - Distribuição de cauda do atraso com o processo de chegada tendo parâmetro de Hurst igual a 0,9	27
Figura 5.1 – Método de Geração por Inversão.....	38
Figura 5.2 – Distribuição Beta	44
Figura 6.1 - Arquitetura do GTAR (extraída de [7]).....	46
Figura 6.2 – Janela: ‘Abrir arquivo’	47
Figura 6.3 – Janela: ‘Importador de dados externos’ – Selecionar coluna.....	47
Figura 6.4 – Janela: ‘Importador de dados externos’ – Selecionar modo de análise dos dados	48
Figura 6.5 – Janela: ‘Inferência’ – Gráfico da distribuição do processo	49
Figura 7.1 – Histograma do tráfego de Poisson gerado.....	50
Figura 7.2 – Tráfego de um backbone.....	52
Figura 7.3 – Tráfego de um backbone referente a app1	53
Figura 7.4 – Tráfego de um backbone referente a app2	53
Figura 7.5 – Tráfego do backbone após aplicação do método de remoção de <i>trend</i>	54
Figura 7.6 – Tráfego do backbone por 1 dia	54

Figura 7.7 – Tráfego do backbone por 1 dia depois da remoção do <i>trend</i>	54
Figura 7.8 – Distribuição de cauda do atraso em um nó para o tráfego Poisson	57
Figura 7.9 – Distribuição de cauda do atraso do tráfego auto-similar com parâmetro de Hurst igual a 0,5.....	57
Figura 7.9 – Distribuição de cauda do atraso do tráfego auto-similar com parâmetro de Hurst igual a 0,7.....	58
Figura 7.10 – Distribuição de cauda do atraso do tráfego auto-similar com parâmetro de Hurst igual a 0,9.....	58
Figura B.1 – Modelo Rajada para uma única fonte de voz.	70
Figura C.1 – Dois nós com igual capacidade de transmissão. Se o tamanho dos pacotes é igual não há espera na fila do segundo nó.	78
Figura C.2 – Diagrama dos tempos de chegada e do tempo gasto na transmissão dos pacotes.	79
Figura C.3 – Exemplo de uma rede de dados.....	80
Figura D.1 – Gráfico de $\phi(t)$	85
Figura D.2 – Gráficos de $\phi_{jk}(t)$	85
Figura D.3 – Wavelet de Haar.....	86
Figura D.4 – Gráficos de $\Psi_{jk}(t)$	86

Lista de Tabelas

Tabela 7.1 – Estimação de parâmetros dos tráfegos fGn gerados	51
Tabela 7.2 – Estimação de parâmetros estatísticos do tráfego de backbone	55
Tabela 7.3 – Estimação de parâmetros estatísticos de app1	55
Tabela 7.4 – Estimação de parâmetros estatísticos de app2	56
Tabela B.1 – Possibilidades de modelagem utilizando fontes rajada	71
Tabela C.1 – Notação A/B/X/Y/Z.....	72

1. Introdução

Nos últimos anos, vem sendo cada vez mais freqüente a utilização de redes comutadas por pacote para prover serviços que, em um passado próximo, eram providos somente em redes comutadas por circuito ou utilizando canais dedicados. Tais serviços, como tráfegos de voz e vídeo, quando executados em tempo real, necessitam de qualidade equivalente à oferecida caso estivessem sendo providos sobre redes comutadas por circuitos; porém, essa qualidade muitas vezes não pode ser oferecida. Essa falta de capacidade da rede pode estar associada a um projeto de rede que não leve em consideração a correta distribuição estatística do tráfego que passa pelo núcleo dessa rede.

Assim, é de grande importância, no correto dimensionamento de redes que servirão de suporte para diversos serviços, o conhecimento das distribuições estatísticas e de certas propriedades dessas distribuições que melhor se aproximam de um tráfego real gerado por esses serviços. E, como forma de garantir que serviços dependentes da rede apresentem comportamento compatível com o que foi projetado, podem ser realizados testes que simulem tráfegos que tenham as mesmas características de um tráfego real, podendo, assim, coletar medidas de qualidade de serviço.

A teoria de filas e o processo poissoniano eram utilizados para fazer a modelagem e o estudo do tráfego em uma rede de dados, mas ambos foram abandonados devido a estudos mais recentes [1] que demonstram que tráfegos agregados possuem características incompatíveis com o processo poissoniano. Esses tráfegos possuem auto-similaridade estatística em diferentes escalas de tempo e dependência de longa duração. O parâmetro utilizado para indicar o grau de auto-similaridade de um processo estocástico é chamado parâmetro de Hurst e pode variar de 0,5 a 1 para processos auto-similares. Mais recentemente surgiu o conceito de multifractalidade, uma nova forma de se caracterizar um tráfego real onde, em escalas menores, o tráfego não é exatamente auto-similar, porém ainda é possível a determinação do grau de auto-similaridade entre escalas. Essa variação é dada por um coeficiente, o expoente de Hölder.

Como já visto, é relevante para o correto projeto de uma rede que se conheça, além distribuição do tráfego que a atravessará, algumas propriedades dessa distribuição. No caso de uma rede de dados, as principais distribuições que nos interessam são a distribuição do tempo entre chegada de pacotes em um nó, a distribuição do tamanho dos pacotes que chegam a esse nó e a distribuição do tempo gasto pelo nó para processar cada pacote. E as principais propriedades dos processos são a auto-similaridade e a dependência de longa duração. Porém, em muitos casos, o projetista não conhece distribuição do tráfego para o qual irá desenvolver seu projeto, mas possui acesso a uma rede existente de onde pode extrair informações necessárias. Com essas informações, poderá utilizar métodos de inferência e relacionar o tráfego real a uma distribuição matemática conhecida. Assim, poderá utilizar modelos matemáticos que o orientem no dimensionamento da rede a ser construída.

Após a construção da rede, de acordo com um projeto específico, e antes da comercialização de serviços sobre essa, é necessário que sejam realizados testes para averiguar o seu correto funcionamento. Entretanto, um simples teste de conexão pode não ser suficiente para que a rede seja considerada pronta e ofereça os serviços de acordo com o projeto. Tornam-se necessários testes que simulem o tráfego que a rede terá que suportar quando em funcionamento e a averiguação da qualidade de serviço que esse tráfego está experimentando na rede.

A motivação inicial desse trabalho foi a criação de dois módulos importantes para a ferramenta apresentada em [7]: o módulo de geração de tráfego e o módulo de inferência. Além disso, esse trabalho visa auxiliar o estudo e desenvolvimento de sistemas de caracterização e controle de tráfego para otimização de recursos em redes multiserviços. Esse estudo está sendo realizado por um grupo de estudos da Universidade de Brasília cuja função principal é a pesquisa de tráfego em redes IP. A proposta de otimização de recursos em redes multiserviços pode ser ilustrada pelo seguinte diagrama.

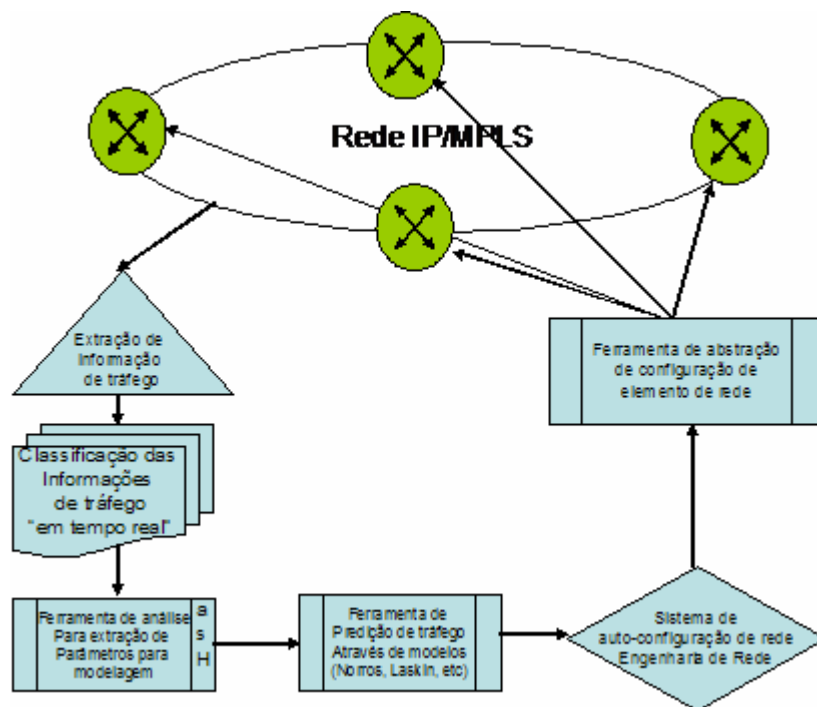


Figura 1.1 – Diagrama de um sistema de otimização de rede

A principal contribuição que esse trabalho pode dar a um sistema de otimização de rede como o ilustrado na figura 1.1 consiste justamente nos módulos que dizem respeito à análise de tráfego para extração de parâmetros para modelagem e predição de tráfego.

O trabalho, objetivando tratar essas questões abordadas, está dividido da seguinte forma: No capítulo 2, apresentamos as bases teóricas do processo auto-similar, suas propriedades, definições e generalizações para abranger mais processos que não o estritamente auto-similar. No capítulo 3, apresentamos o modelo multifractal que apresenta uma modelagem para o tráfego real para pequenas escalas. No capítulo 4, apresentamos conceitos sobre inferência estatística, explanando os métodos de inferência para que seja possível relacionar os dados obtidos com uma distribuição matemática conhecida (caso exista) e as propriedades dessa distribuição. No capítulo 5, explanamos sobre várias formas de se gerar tráfegos a partir de diversos processos estocásticos. No capítulo 6, apresentamos a base inicial do nosso estudo, a implementação dos módulos de geração de tráfego e de inferência estatística no software de código aberto GTAR apresentado em [7] como uma nova ferramenta para realizar medidas de qualidade de serviço em redes multiserviços. No capítulo 7,

discutimos os resultados obtidos utilizando a ferramenta de geração de tráfego e a ferramenta de inferência, para verificar a aderência dos tráfegos gerados e reais aos modelos matemáticos. No capítulo 8, deixamos nossas conclusões sobre o estudo que realizamos e sobre as implementações que tornaram possível maior capacidade analítica e melhor visualização prática do trabalho realizado. No anexo A, mostramos as características do processo de Poisson, primeiro modelo utilizado para caracterizar um tráfego que é recebido em um nó de uma rede. No anexo B, apresentamos o modelo de tráfego rajada onde uma representa uma fonte de tráfego e as diversas possibilidades que podem ser implementadas utilizando fontes ON/OFF. No anexo C, apresentamos os conceitos sobre a teoria de filas, sobre medidas de eficiência teóricas e sobre o Teorema de Kleinrock. No anexo D, apresentamos o conceito de multiresolução, teoria que é utilizada pelos métodos de geração WIG e MWM, apresentados no capítulo 5.

2. Tráfego Auto-similar

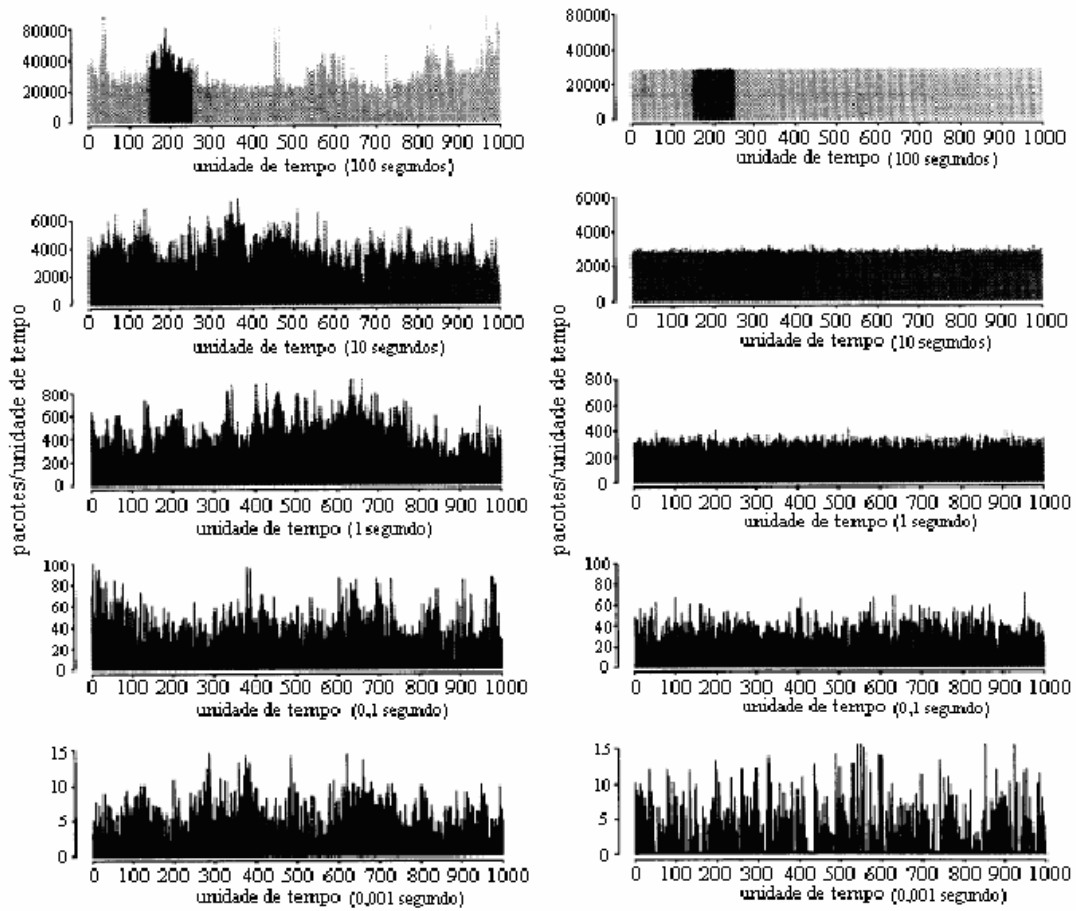
Até 1994, o modelo M/M/1, explicado anteriormente, era utilizado para o dimensionamento de redes de dado. No entanto, a utilização desse modelo passou a ser altamente questionada devido aos resultados obtidos em [1]. Foi feito um estudo sobre as características estatísticas do tráfego em uma rede Ethernet. Esse estudo mostrou que o tráfego dessa rede possuía duas propriedades que não eram compatíveis com o modelo de tráfego até então utilizado: auto-similaridade estatística, característica que o tráfego possui de manter as propriedades estatísticas em diferentes escalas de tempo, e dependência de longa duração.

Podemos facilmente perceber pela figura abaixo que, à medida que aumentamos a escala de tempo, as propriedades estatísticas do tráfego poissoniano se degeneram muito mais rapidamente que as do tráfego real estudado. Em particular, podemos perceber visualmente que a variância do tráfego poissoniano se degenerou muito mais que a do tráfego real, já que para escalas grandes de tempo o tráfego Poissoniano se mantém praticamente constante, enquanto o tráfego real ainda possui um grande número de “picos” de tráfego.

Outro argumento forte para a utilização de modelos auto-similares de tráfego é o fato de que, em muitos casos, o tráfego de uma rede pode ser visto como uma superposição de várias fontes independentes ON/OFF que possuam uma distribuição de cauda pesada para o período ON. Pode ser provado que a agregação de tais fontes convergem para um processo gaussiano auto-similar [2].

Essa característica de invariância à escala pode ser útil para que tenhamos uma noção melhor de como caracterizar esse efeito de *burstiness* dos tráfegos. Esse efeito é importante, pois níveis altos de *burstiness* levam a uma menor utilização de recursos de rede para uma mesma qualidade de serviço e possui direto impacto na administração da rede.

Nesse capítulo definiremos matematicamente a auto-similaridade estatística e algumas de suas propriedades, mostrando sua relação com a dependência de longa duração.



**Figura 2.1 – Comparação entre um tráfego poissoniano e um tráfego real
(extraída de [6])**

2.1. Auto-Similaridade

Como já vimos, o segmento de um tráfego auto-similar medida a uma escala de tempo tem as mesmas características estatísticas que uma versão apropriadamente escalada do tráfego a uma escala de tempo diferente. Vamos agora definir matematicamente a auto-similaridade para um processo estocástico em tempo contínuo.

Um processo em tempo contínuo $X(t)$ é considerado exatamente auto-similar se:

$$X(t) \stackrel{d}{=} a^{-H} X(at), a > 0 \quad (2.1)$$

em que a igualdade possui sentido probabilístico. Isso quer dizer que as funções amostras de um processo $X(t)$ e as de sua versão escalada $a^{-H}X(at)$, obtidas comprimindo-se o eixo do tempo por fator a e o eixo das amplitudes por um fator a^H , não podem ser distinguidas estatisticamente. Ou seja, os momentos de ordem n de $X(t)$ são iguais aos momentos de ordem n de $X(at)$ escalados por a^{-Hn} . Caso o processo tenha as características de auto-similaridade para as estatísticas de segunda ordem ele é chamado de exatamente auto-similar de segunda ordem e seus momentos de primeira e segunda ordem são dados por:

$$E[X(t)] = \frac{E[X(at)]}{a^{-H}} \quad (2.2)$$

$$Var[X(t)] = \frac{Var[X(at)]}{a^{-2H}} \quad (2.3)$$

$$R_X(t_1, t_2) = \frac{R_X(at_1, at_2)}{a^{-2H}} \quad (2.4)$$

Percebemos que o parâmetro de Hurst (H) indica o grau de auto-similaridade do processo. Quanto mais próximo de 1 for H mais auto-similar será o processo.

Uma consequência direta de (2.1) é que todos os momentos de ordem n de $X(t)$, caso existam, se comportam de acordo com uma lei de potência em função do tempo:

$$E|X(t)|^n = E|X(1)|^n |t|^{nH} \quad (2.5)$$

Modelos práticos de processos auto-similares geralmente são restringidos para os processos auto-similares com incrementos estacionários (processos H-sssi). Ou seja, seu processo de incrementos $Y(\delta, t)$ é tal que:

$$\{Y(\delta, t) = Y_\delta(t) = X(t + \delta) - X(t), t \in R\} \stackrel{d}{=} \{X(\delta) - X(0)\}, \forall \delta \quad (2.6)$$

Para um processo H-sssi, o parâmetro de Hurst está necessariamente entre o intervalo: $0 < H < 1$. Além disso, sua função de autocorrelação, caso exista, é da seguinte forma:

$$R_X(t_1, t_2) = \frac{E|X(1)|^2}{2} \left(|t_1|^{2H} + |t_2|^{2H} - |t_2 - t_1|^{2H} \right) \quad (2.7)$$

A equação (2.7) pode ser provada da seguinte forma:

$$\begin{aligned} E|X(t_2) - X(t_1)|^2 &= E|X(t_2)|^2 + E|X(t_1)|^2 - 2E|X(t_1)||X(t_2)| \\ E|X(t_1)||X(t_2)| &= R_X(t_1, t_2) = \frac{1}{2} \left\{ E|X(t_1)|^2 + E|X(t_2)|^2 - E|X(t_2) - X(t_1)|^2 \right\} \end{aligned}$$

Como $X(t)$ possui incrementos estacionários:

$$E|X(t_2) - X(t_1)|^2 = E|X(t_2 - t_1)|^2$$

Então a função de autocorrelação de $X(t)$ é dada por:

$$\begin{aligned} R_X(t_1, t_2) &= \frac{1}{2} \left\{ E|X(1)|^2 |t_1|^{2H} + E|X(1)|^2 |t_2|^{2H} - E|X(1)|^2 |t_2 - t_1|^{2H} \right\} \\ R_X(t_1, t_2) &= \frac{E|X(1)|^2}{2} \left\{ |t_1|^{2H} + |t_2|^{2H} - |t_2 - t_1|^{2H} \right\} \end{aligned}$$

Além disso, a função de autocorrelação do processo de incrementos de $X(t)$ é:

$$R_{Y_\delta}(\tau) = \frac{E|X(1)|^2}{2} \left\{ |\tau + \delta|^{2H} + |\tau - \delta|^{2H} - 2|\tau|^{2H} \right\} \quad (2.8)$$

Isso é facilmente demonstrado da seguinte forma:

$$\begin{aligned} R_{Y_\delta}(\tau) &= E|Y_\delta(t)||Y_\delta(t + \tau)| = E[(X(t + \delta) - X(t))(X(t + s + \delta) - X(t + s))] \\ R_{Y_\delta}(\tau) &= E[X(t + \delta)X(t + s + \delta)] - E[X(t + \delta)X(t + s)] - E[X(t)X(t + s + \delta)] + \\ &\quad + E[X(t)X(t + s)] \\ R_{Y_\delta}(\tau) &= \frac{E|X(1)|^2}{2} \left\{ |t + \delta|^{2H} + |t + s + \delta|^{2H} - |s|^{2H} - |t + \delta|^{2H} - |t + s|^{2H} + |s - \delta|^{2H} + \right. \\ &\quad \left. - |t|^{2H} - |t + s + \delta|^{2H} + |s + \delta|^{2H} + |t|^{2H} + |t + s|^{2H} - |s|^{2H} \right\} \\ R_{Y_\delta}(\tau) &= \frac{E|X(1)|^2}{2} \left\{ |\tau + \delta|^{2H} + |\tau - \delta|^{2H} - 2|\tau|^{2H} \right\} \end{aligned}$$

A auto-similaridade de $X(t)$ também é passada para seus incrementos de forma que:

$$Y(\delta, t) \stackrel{d}{=} a^{-H} Y(a\delta, at) \quad (2.9)$$

$$E|Y(\delta, t)|^n = E|X(t + \delta) - X(t)|^n = E|X(1)|^n |\delta|^{nH} \quad (2.10)$$

2.2. Dependência de Longa Duração

Um processo é dito ter dependência de longa duração quando sua função de autocorrelação $r_X(\tau)$ decai tão devagar que a sua soma diverge, ou seja, para qualquer $A > 0$:

$$\int_A^\infty R_X(\tau) d\tau = +\infty$$

Ou equivalentemente para processos em tempo discreto:

$$\sum_k r_X(k) = \infty$$

Geralmente os processos com dependência de longa duração possuem função de autocorrelação que decaem lentamente de acordo com uma função hiperbólica, ou seja, funções do tipo:

$$r_X(\tau) \sim c_1 |\tau|^{\gamma-1}, \tau \rightarrow \infty, \gamma \in (0,1) \quad (2.11)$$

Processos com essa função de autocorrelação possuem a seguinte densidade espectral de potência:

$$f_X(\lambda) \sim c_2 |\lambda|^{-\gamma}, \lambda \rightarrow 0, \gamma \in (0,1) \quad (2.12)$$

Qualquer processo exatamente auto-similar possui dependência de longa duração. De fato, como provado em [18], a função de autocorrelação de $Y_\delta(t)$ é dada assintoticamente por:

$$r_{Y_\delta}(\tau) \sim c\tau^{2(H-1)}, \tau \gg \delta \quad (2.13)$$

Comparando as equações (2.11) e (2.13), vemos que o processo de incrementos possui longa duração quando $\frac{1}{2} < H < 1$.

2.3. Auto-Similaridade em Tempo Discreto

Definimos anteriormente a auto-similaridade para processos estocásticos em tempo contínuo e verificamos algumas de suas propriedades. Em particular, verificamos as propriedades estatísticas de até segunda ordem para um processo auto-similar estacionário: $Y(\delta, t)$. Se definirmos um processo em tempo discreto a partir de $Y(\delta, t)$, para um valor fixo de lag δ , como sendo:

$$G(k) = Y(k\delta) = X((k+1)\delta) - X(k\delta) \quad (2.14)$$

A partir de (2.8), podemos derivar a função de autocorrelação do processo estacionário exatamente auto-similar de segunda ordem em tempo discreto definido em (2.14) da seguinte forma:

$$\begin{aligned} r_G(k) &= E[G(n)G(n+k)] = E[G(n\delta)G(n\delta+k\delta)] \\ r_G(k) &= \frac{E[X(1)]^2}{2} \{ |k\delta + \delta|^{2H} + |k\delta - \delta|^{2H} - 2|k\delta|^{2H} \} \\ r_G(k) &= \frac{E[X(1)]^2 |\delta|^{2H}}{2} [|k+1|^{2H} + |k-1|^{2H} - 2|k|^{2H}] \\ r_G(k) &= \frac{E[G(n)]^2}{2} [|k+1|^{2H} + |k-1|^{2H} - 2|k|^{2H}] \end{aligned} \quad (2.15)$$

Além disso, pode ser provado que o coeficiente de autocorrelação do processo $G(k)$ é dado por:

$$\rho_G(k) = \frac{E[(X(n) - \mu)(X(n+k) - \mu)]}{E[(X(n) - \mu)]^2} = \frac{1}{2} [|k+1|^{2H} + |k-1|^{2H} - 2|k|^{2H}]$$

Um processo em tempo discreto é exatamente auto-similar quando seu coeficiente de autocorrelação é dado por:

$$\rho(k) = \rho^{(m)}(k) = \frac{1}{2} \left[|k+1|^{2H} + |k-1|^{2H} - 2|k|^{2H} \right] \forall m, k \quad (2.16)$$

em que $\rho^{(m)}(k)$ é o coeficiente de autocorrelação do processo m-agregado de $G(k)$, definido em (2.17).

O processo m-agregado de um processo $G(k)$ é definido da seguinte forma:

$$G^{(m)}(k) = \frac{1}{m} \sum_{i=km}^{(k+1)m-1} G(i) \quad (2.17)$$

Ou seja, cada valor de $G^{(m)}(k)$ é formado pela média dos m valores consecutivos de $G(k)$, a partir de $m \cdot k$:

$$G^{(m)}(k) = \frac{1}{m} [G(km) + G(km+1) + G(km+2) + \dots + G(km+m-1)]$$

A dependência de longa duração dos processos auto-similares faz com que a variância de seus processos m-agregados decaia muito mais lentamente que a variância de processos com dependência de curta duração. Para o caso do processo $G(k)$ definido em (2.14) temos:

$$\text{var}(G^{(m)}(0)) = \text{var}\left(\frac{1}{m} \sum_{i=0}^{m-1} G(i)\right) = \text{var}\left(\frac{1}{m} \sum_{i=0}^{m-1} (X((i+1)\delta) - X(i\delta))\right) = \text{var}\left(\frac{1}{m} (X(m\delta) - X(0))\right)$$

Como $X(0) = 0$, então:

$$\text{var}(G^{(m)}(0)) = \text{var}\left(\frac{1}{m} X(m\delta)\right) = \text{var}\left(\frac{m^H}{m} X(\delta)\right) = \text{var}(m^{H-1} G(0)) = m^{2H-2} G(0)$$

Já que o processo $G(k)$ é estacionário, a variância de $G^{(m)}(k)$ para qualquer k é da forma:

$$\text{var}(G^{(m)}(k)) \sim m^{2H-2} \quad (2.18)$$

Uma das principais vantagens do modelo auto-similar está na sua simplicidade de caracterização: todas as suas propriedades podem ser definidas e controladas pelo

parâmetro de Hurst. No entanto, essa simplicidade é também o principal problema dos modelos de tráfego auto-similar. Essa definição rígida de auto-similaridade dificilmente é atendida por processos reais. Em primeiro lugar, a definição (2.1) é válida para qualquer α real e positivo. Isso implica que a auto-similaridade existe para qualquer escala e que a relação entre os fatores de escala se mantém a mesma para todas as escalas. Em tráfegos reais, no entanto, a auto-similaridade só existe para um número finito de escalas e a relação entre os fatores de escala se mantém apenas aproximadamente iguais para as diferentes escalas. De fato, os tráfegos reais são, tipicamente, assintoticamente auto-similar, ou seja, apresentam características auto-similares apenas para escalas muito grandes. Além disso, a auto-similaridade implica, como pode ser visto pela equação (2.5), que todos os momentos de ordem q (caso exista) são controlados por um único parâmetro H , o que não acontece com dados empíricos. No caso da modelagem de tráfegos, na maior parte das vezes, modelos de auto-similaridade com apenas um parâmetro são válidos para grandes escalas enquanto mais parâmetros são necessários para modelagem em pequenas escalas.

A primeira generalização de um processo exatamente auto-similar pode ser feita considerando que o processo será exatamente auto-similar de segunda ordem quando a relação (2.1) é válida para as estatísticas de até segunda ordem. Essa definição ainda é muito estrita para ser usada como modelo de processos reais, pois generaliza a auto-similaridade apenas no que diz respeito aos momentos de ordem n . Por esse modelo todas as estatísticas de até segunda ordem (como variância, função de autocorrelação e coeficiente de autocorrelação) ainda são completamente determinadas pelo parâmetro de Hurst em todas as escalas.

Outra possível generalização é considerarmos o processo com dependência de longa duração definido na seção 2.2. Essa definição generaliza a função de autocorrelação de um processo auto-similar. Como vimos, a função de autocorrelação de um processo auto-similar exibe, assintoticamente, exatamente o mesmo comportamento da função de autocorrelação de um processo com dependência de longa duração definido por (2.11). Ou seja, a relação (2.15) não precisa ser válida para todas as escalas, basta que ela seja válida assintoticamente (para escalas grandes).

Outros dois modelos que generalizam o modelo auto-similar no que diz respeito à escala são o processo forte assintoticamente auto-similar de segunda ordem e o processo assintoticamente auto-similar de segunda ordem.

Um processo é chamado de forte assintoticamente auto-similar de segunda ordem quando as variâncias de seus processos agregados são da seguinte forma:

$$\lim_{m \rightarrow \infty} \text{var}(X^{(m)}) \sim m^{-\beta}, \text{ com } H = 1 - \frac{\beta}{2} \quad (2.19)$$

O processo é chamado de assintoticamente auto-similar de segunda ordem quando a relação (2.16) é válida apenas assintoticamente:

$$\lim_{m \rightarrow \infty} \rho^{(m)}(k) = \frac{1}{2} \left[|k+1|^{2H} + |k-1|^{2H} - 2|k|^{2H} \right] \quad (2.20)$$

A relação entre processos exatamente auto-similares, assintoticamente auto-similares, forte assintoticamente auto-similares e com dependência de longa duração está demonstrada na figura 2.2, como provado em [18].

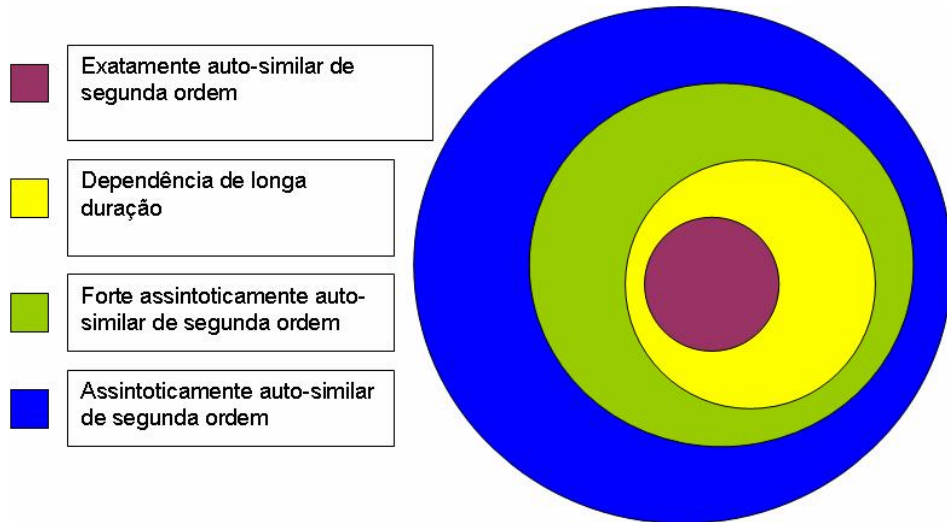


Figura 2.2 – Relação entre diferentes tipos de processos

Como podemos perceber pela figura 2.2, dentre as generalizações estabelecidas para um processo auto-similar, o processo exatamente auto-similar de segunda ordem é o mais estrito e o assintoticamente auto-similar de segunda ordem é o mais geral.

De acordo com o que foi visto, os modelos que podem ser usados para caracterizar um tráfego real, definem apenas o comportamento estatístico dos tráfegos para grandes escalas. No entanto, a correlação do processo em escalas pequenas também podem influenciar na resposta de um sistema a esse tráfego. Com o intuito de se modelar o comportamento do tráfego também para escalas pequenas foi proposto o tráfego multifractal.

2.4. Movimento Browniano Fracionário (fBm) e Ruído Gaussiano Fracionário (fGn)

O Movimento Browniano fracionário é o mais usado modelo de processo auto-similar. Isso se deve ao fato dele ser o único processo auto-similar gaussiano que possui incrementos estacionários. Além disso, como provado em [2], a superposição de várias fontes ON/OFF com a distribuição do período ON sendo de cauda pesada resulta, pelo teorema do limite central, em um processo auto-similar gaussiano. O processo de incrementos do fBm é o ruído Gaussiano fracionário.

Como o fBm é um processo auto-similar com incrementos estacionários (H-sssi), ele e seu processo de incrementos, fGn, possuem todas as características derivadas anteriormente para um processo H-sssi e para seu processo de incrementos respectivamente.

3. Tráfego Multifractal

Como mencionado anteriormente, a modelagem de um tráfego real por um tráfego auto-similar controlado apenas pelo parâmetro de Hurst geralmente é válido para escalas grandes. No entanto, para escalas pequenas ele normalmente não é muito adequado. Veremos nessa seção, como o parâmetro de Hurst influencia o tráfego em pequenas escalas e definiremos um modelo mais complexo para o comportamento do tráfego em pequenas escalas.

Para analisarmos a influência do parâmetro de Hurst em pequenas escalas é necessário introduzirmos o conceito de regularidade de Hölder pontual. A regularidade de Hölder pontual descreve a regularidade das funções amostras de um processo estocástico por meio de uma comparação local com uma função que siga uma lei de potência. Um sinal $X(t)$ é dito ter regularidade de Hölder $h \geq 0$ em t_0 se for possível encontrar um polinômio de grau $n = \lfloor h \rfloor$ e uma constante $K > 0$ tal que:

$$|X(t) - P_n(t - t_0)| \leq K|t - t_0|^h \quad (3.1)$$

No caso em que $0 \leq h < 1$, a parte regular de $X(t)$ se reduz a $P_n(t - t_0) = X(t_0)$, simplificando a relação (3.1) em uma relação baseada apenas em incrementos da forma:

$$|X(t_0 + \delta) - X(t_0)| \leq K|\delta|^h \quad (3.2)$$

O maior valor de h é chamado de expoente Hölder. O expoente de Hölder indica a regularidade local do processo. Um processo que possua expoente de Hölder constante em toda a função amostra, para qualquer t_0 , é chamado de monofractal.

O expoente de Hölder generaliza a noção de diferenciabilidade, e serve para fornecer uma medida da regularidade da função amostra na vizinhança. Se o expoente de Hölder estiver entre 0 e 1, o processo é contínuo e não diferenciável em t_0 . O expoente de Hölder é utilizado para caracterizar a singularidade. Quanto menor for o expoente de Hölder (entre 1 e 0) maior é a variação local das funções amostras. Para o

caso de um tráfego de rede, um expoente de Hölder menor em t_0 significa um surto de tráfego com mais intensidade na região de t_0 .

Muitas vezes a regularidade de Hölder é estudada através do processo de incrementos de acordo com a seguinte relação:

$$E[X(t + \delta) - X(t)]^2 \sim \sigma^2 |\delta|^{2h}, \delta \rightarrow 0 \quad (3.3)$$

De fato, o comportamento descrito em (3.3) para um processo com incrementos estacionários é fortemente relacionado com a regularidade de Hölder.

Para o caso de um processo auto-similar como o fBm, pode ser mostrado que a relação (3.2) é válida para qualquer $h < H$ em praticamente todas as funções amostras para qualquer valor de t_0 . Confirmando que o processo auto-similar possui uma característica monofractal, controlada por H . Isso significa que o processo possui oscilações com igual força durante todo o tempo.

Apesar de uma prova rigorosa ser difícil, podemos ter uma forte intuição da característica monofractal de um processo fBm. Como já foi visto, um processo auto-similar com incrementos estacionários (H-sssi) possui incrementos com valor quadrático médio dado por:

$$E[Y(\delta, t)]^2 = E[X(t + \delta) - X(t)]^2 = E[X(1)]^2 |\delta|^{2H}$$

Supondo que $X(t)$ seja Gaussiano, ou seja, restringindo o processo $X(t)$ para o movimento Browniano fracionário e $Y(t)$ para o ruído Gaussiano fracionário, temos que a maior parte das amostras está a poucos desvios padrões da média. Isso quer dizer que as oscilações de X em intervalos de tamanho δ , serão próximas de $\delta^H \sqrt{E[X(1)]^2}$ para toda a função amostra, independente de t .

Podemos conseguir um modelo mais geral permitindo que h em (3.3) seja uma função variante no tempo $h(t)$. Um processo desse tipo pode descrever um processo que possui, localmente, características de um fractal que variam ao longo do tempo. Caso $0 < h(t) < 1$ seja uma função determinística que varia com regularidade, então o processo

X é chamado de “multifractal”, ou, se X for também Gaussiano, ele é chamado de localmente auto-similar. Apesar de esse modelo ser localmente fractal com um expoente que varia ao longo do tempo, ele não pode ser chamado de multifractal por dois motivos: $h(t)$ é determinístico, ou seja, para um determinado tempo t_0 a regularidade de Hölder pontual é sempre a mesma para qualquer função amostra. No entanto, $h(t)$ é aleatório para um processo multifractal. Além disso, $h(t)$ varia regularmente, mas em processos multifractais $h(t)$ varia irregularmente, podendo assumir valores muito distantes em qualquer intervalo de tempo.

Muitas vezes, tráfegos reais na rede possuem características de multifractais. Isso quer dizer que o expoente de Hölder do processo, definido para cada função amostra como sendo o maior h tal que (3.1) é verdadeiro, é uma variável aleatória e varia erráticamente para cada t_0 . Ou seja, a regularidade $h(t)$ de um processo multifractal também é um processo estocástico.

Para os processos multifractais, as variações da regularidade não são descritos por uma função $h(t)$, mas pelo chamado espectro multifractal. O espectro multifractal $D(h)$ fornece informação em relação a quantidade de pontos que possuem expoente de Hölder $h(t) = h$. O expoente de Hölder pode ser definido de uma função amostra de $X(t)$ da seguinte forma:

$$h(t) = \frac{\log|X(t + \delta) - X(t)|}{\log|\delta|}, \delta \rightarrow 0 \quad (3.4)$$

Alternativamente, se dividirmos $X(t)$ em 2^n intervalos de tamanho 2^{-n} , o expoente de Hölder pode ser definido alternativamente pela equação (3.5). Para facilitarmos a notação definimos que $k_n 2^{-n} \rightarrow t$ quer dizer que $t \in [k_n 2^{-n}, (k_n + 1) 2^{-n})$, com $n \rightarrow \infty$. Logo, o expoente de Hölder pode ser definido da seguinte forma:

$$h(t) = \lim_{k_n 2^{-n} \rightarrow t} \frac{\log|X((k_n + 1) 2^{-n}) - X(k_n 2^{-n})|}{\log|(k_n + 1) 2^{-n} - k_n 2^{-n}|} = \lim_{k_n 2^{-n} \rightarrow t} \frac{\log|X((k_n + 1) 2^{-n}) - X(k_n 2^{-n})|}{\log|2^{-n}|}$$

$$h(t) = \lim_{k_n 2^{-n} \rightarrow t} -\frac{1}{n} \log_2 |X((k_n + 1) 2^{-n}) - X(k_n 2^{-n})|$$

$$h(t) = \lim_{k_n 2^{-n} \rightarrow t} a_{k_n}^n \text{ em que } a_{k_n}^n = -\frac{1}{n} \log_2 |X((k_n + 1)2^{-n}) - X(k_n 2^{-n})| \quad (3.5)$$

Baseado na definição (3.5), que considera a divisão de $X(t)$ em 2^n intervalos de tamanho 2^{-n} , podemos definir matematicamente o espectro multifractal:

$$D(h) = \lim_{\varepsilon \rightarrow 0} \lim_{n \rightarrow \infty} \frac{1}{n} \log_2 m \quad (3.6)$$

em que m representa a quantidade dos 2^n intervalos que possuem expoente de holder aproximadamente igual a h , ou seja, cujos $a_{k_n}^n \in (h - \varepsilon, h + \varepsilon)$.

Por definição, temos que $0 < D(h) < 1$. Para um processo monofractal, em que $h(t) = H$ para todo t , temos que $D(H) = 1$ e $D(h) = 0$, $\forall h \neq H$. Para processos multifractais o espectro multifractal, definido em (3.6) para cada função amostra, é aleatório.

Muitas vezes, é mais fácil derivar o espectro multifractal a partir das funções de partição, definidas como:

$$\tau(q) = \lim_{\delta \rightarrow 0} \frac{\log S_\delta(q)}{\log \delta}, \text{ com } S_\delta(q) = \sum_k |X((k+1)\delta) - X(k\delta)|^q \quad (3.7)$$

Ou analogamente:

$$\tau(q) = \lim_{n \rightarrow \infty} -\frac{1}{n} \log_2 E[S_n(q)], \text{ com } S_n(q) = \sum_{k_n} |X((k_n + 1)2^{-n}) - X(k_n 2^{-n})|^q \quad (3.8)$$

Pode ser provado [16] que o espectro multifractal se relaciona com as funções de partição a partir da transformada de Legendre da seguinte forma:

$$\tau(q) = D^*(q) = \inf_{h \in R} (qh - D(h)) \quad (3.9)$$

Se $\tau(q)$ existir e for diferenciável pode ser provado [ref] que o dual também é válido. Ou, seja:

$$D(h) = \tau^*(h) = \inf_q (qh - \tau(q)) \quad (3.10)$$

Essa relação é chamada de formalismo Multifractal. A resolução da equação (3.10) pode ser dada por (3.11):

$$\begin{aligned} \text{Caso } \tau''(q) < 0: \\ \tau^*(h) = qh - \tau(q), \text{ com } h = \tau'(q) \end{aligned} \quad (3.11)$$

A partir das relações dadas pelo formalismo multifractal podemos derivar o espectro multifractal de um processo auto-similar com incrementos estacionários. Primeiro derivamos a função de partição de um processo auto-similar:

$$E[S_n(q)] = E \left[\sum_{k_n=0}^{2^n-1} |X((k_n+1)2^{-n}) - X(k_n 2^{-n})|^q \right]$$

Já que X(t) possui incremento estacionários, temos que:

$$E[S_n(q)] = 2^n E \left[|X(2^{-n})|^q \right] = 2^n 2^{-nqH} E \left[|X(1)|^q \right] = 2^{-n(qH-1)} E \left[|X(1)|^q \right]$$

$$\tau(q) = \lim_{n \rightarrow \infty} -\frac{1}{n} \ln E[S_n(q)] = \begin{cases} qH - 1 & \text{para } q > -1, \\ -\infty & \text{para } q \leq -1 \end{cases}$$

Pelo formalismo multifractal, o espectro multifractal é dado por:

$$D(h) = \tau^*(q)$$

cuja resolução é dada por:

$$D(h) = qH - \tau(q) = qH - (qH - 1) = 1, \text{ com } h = \tau'(q) = H$$

Ou seja, como esperado, o processo auto-similar é monofractal e possui expoente de Hölder H para qualquer ponto da função amostra, independente de t.

3.1. Cascatas Multiplicativas

Uma classe de modelos que possuem características ricas suficientes para modelar as propriedades de um processo multifractal são as cascatas multiplicativas. A construção de uma cascata multiplicativa é baseada na criação iterativa dos detalhes do processo em escalas mais finas, a partir de escalas mais amplas, de forma multiplicativa.

Uma das cascatas multiplicativas mais usadas é a cascata binomial. A cascata binomial pode ser definida para um intervalo de tamanho N da seguinte forma:

$$\begin{aligned} X((2k+1)/2^{n+1}) - X((2k)/2^{n+1}) &= M_{2k}^{n+1} \cdot (X((k+1)/2^n) - X(k/2^n)) \\ &= \prod_{i=1}^{n+1} M_{k_i}^i (X(N) - X(0)) \end{aligned} \quad (3.12)$$

Os multiplicadores M_{ki}^i são variáveis aleatórias independentes positivas e a soma dos multiplicadores “irmãos” é 1: $M_{2k}^{n+1} + M_{2k+1}^{n+1} = 1$. A cascata binomial pode ser construída iterativamente da seguinte forma. Um intervalo de tamanho T que possua valor M_k^n é dividido na metade em dois intervalos de tamanho T/2 que possuem valores $M_{2k}^{n+1}M_k^n$ e $M_{2k+1}^{n+1}M_k^n$. Como a soma dos multiplicadores “irmãos” é 1, a soma dos valores dos dois intervalos de tamanho T/2 na escala menor é igual ao valor do intervalo de tamanho T na escala anterior. A figura 3.1 ilustra as três primeiras etapas da construção de uma cascata binomial e a figura 3.2 mostra uma função amostra criada utilizando esse método.

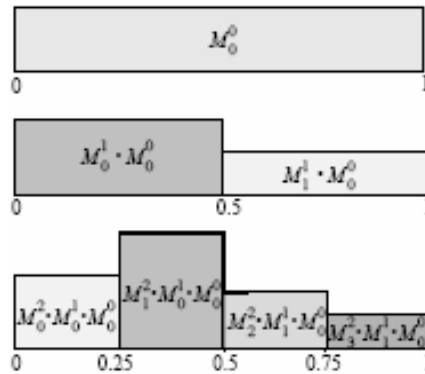


Figura 3.1 – Construção de uma cascata binomial

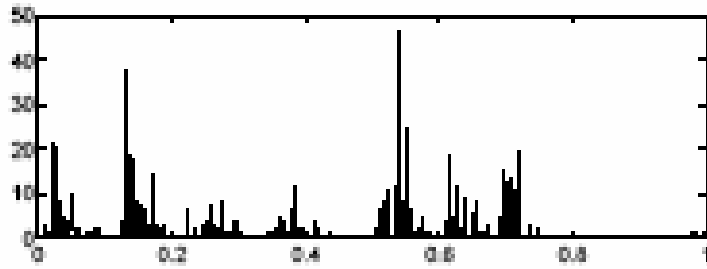


Figura 3.2 – Função amostra criada utilizando-se o método da cascata binomial

Podemos ter uma intuição bem forte da multifractalidade das cascatas multiplicativas a partir de (3.12). Já que os $M_{k_i}^i$ são variáveis aleatórias positivas independentes, temos que, diferentemente de um processo monofractal, a relação entre escalas diferentes em um processo vai depender das $M_{k_i}^i$ e será diferente para diferentes pontos do processo. De fato, a forma do espectro multifractal para uma cascata multiplicativa possui, tipicamente, a forma \cap , enquanto que para processos auto-similares, como mostrado anteriormente, é formado por apenas um ponto em H ($D(H)=1$).

4. Inferência

A maior parte dos sistemas reais possui uma ou mais fontes de aleatoriedade. Tipicamente essas fontes são representadas pela sua esperança ou pela distribuição de probabilidade. Muitas vezes, a representação da aleatoriedade em um sistema por sua esperança pode ser enganosa (a não ser que sua variância seja muito pequena) e normalmente é usada apenas quando não é possível extrair informações suficientes da fonte de aleatoriedade. O problema de se modelar fontes de aleatoriedade por sua média pode ser pelo seguinte exemplo.

Suponha uma fila M/M/1 cujo tempo médio entre duas chegadas consecutivas seja de 0,5 segundos e cujo tempo médio de atendimento seja de 0,4 segundos. Nesse caso, o atraso médio do sistema (tempo médio que cada usuário demora a ser atendido) seria de 2 segundos. Se modelássemos as fontes de aleatoriedade desse sistema por suas respectivas médias, ou seja, se os tempos entre suas chegadas fossem sempre exatamente 0,5 segundos e os tempos de atendimento fossem sempre 0,4 segundos, então cada usuário seria imediatamente atendido e o atraso médio do sistema seria dado somente pelo tempo de atendimento (0,4 segundos).

Além disso, caso não se escolha a distribuição correta para modelagem das fontes de aleatoriedade, a precisão do modelo também pode ser afetada. Isso pode ser mostrado da seguinte forma.

Fizemos duas simulações de filas que possuam um servidor e com tempo médio entre as chegadas de 0,5 segundos. Na primeira simulação fizemos o tempo entre as chegadas com distribuição exponencial e na segunda com distribuição de Rayleigh. Em ambos os casos, fizemos a distribuição do tempo de atendimento como sendo exponencial com média de 0,4 segundos. Em cada simulação utilizamos um milhão de amostras. Na primeira obtivemos um atraso médio no sistema de aproximadamente 2,003 segundos e na segunda obtivemos um atraso médio de aproximadamente 1,324 segundos.

Caso seja possível coletar dados do processo estocástico que se deseja modelar, podemos utilizá-los para simularmos o processo estocástico das seguintes formas:

1. Utilizar os próprios valores dos dados coletados na simulação.
2. Utilizar os próprios valores para definir uma função de distribuição empírica e gerarmos os valores da simulação de acordo com essa distribuição.
3. Utilizar técnicas de inferência estatística para estimar uma forma de distribuição teórica, como a distribuição exponencial. Os valores da simulação seriam então gerados a partir dessa distribuição teórica

Se for utilizado o método 1, a simulação só poderá reproduzir o que já aconteceu historicamente no sistema e muitas vezes não é possível obter dados suficientes para todas as simulações desejadas. Quando se utiliza o método 2, pode-se, por exemplo, utilizar o método de geração por inversão para gerar valores i.i.d (independentes e identicamente distribuídos) com a distribuição empírica dos dados obtidos. Dessa forma, quaisquer valores entre o mínimo valor observado e o máximo valor observado pode ser obtido, evitando os problemas do método 1. Por isso, o método 2 geralmente é preferível ao método 1.

Caso seja possível achar uma distribuição teórica que modele os dados observados, o método 3 é preferível aos outros 2 métodos pelas seguintes razões:

- A distribuição teórica obtida a partir dos dados observados pode ter algumas irregularidades, principalmente se a quantidade de dados for pequena. Uma distribuição teórica ameniza essas irregularidades e podem fornecer mais informações sobre o processo estocástico, a partir de parâmetros dessa distribuição. Dessa forma, pode-se gerar valores a partir dessa distribuição modificando seus parâmetros de forma a estimar o comportamento do sistema em diferentes situações. Por exemplo, caso se determine que os tempos entre chegadas em uma fila se adaptam a uma distribuição exponencial com determinada média, pode-se estimar como o sistema se comportaria caso o tempo médio entre as chegadas fosse menor. Assim, seria possível modelar o sistema para o caso de uma taxa de utilização maior.

- Utilizando-se o método 2 para gerar valores a partir do método da geração por inversão, não é possível gerar valores fora do intervalo dos dados observados. No entanto, algumas vezes a performance do sistema depende da probabilidade de ocorrência de eventos “extremos” que podem estar fora do intervalo de dados observados.
- Algumas vezes existem razões físicas fortes para o uso de determinada distribuição na modelagem de determinado processo estocástico.
- Uma distribuição teórica é uma forma muito mais compacta de se representar um conjunto de dados que uma distribuição empírica. Normalmente, caso a distribuição for estimada a partir de n valores, é necessário armazenar $2n$ valores para representação da distribuição empírica, enquanto que para uma distribuição teórica basta sua p.d.f, definida por poucos parâmetros.

Além da distribuição, podem ser simuladas outras características estatísticas do processo que influenciam no desempenho do processo, como a auto-similaridade e a dependência entre as amostras. Essa influência pode ser ilustrada pelas seguintes simulações.

Simulamos 5 filas com o processo de chegadas sendo um processo auto-similar Gaussiano (fBm). Nas 5 simulações estabelecemos uma média de 10 pacotes por 0,1 segundos com variância de 10, variando apenas o parâmetro de Hurst. Os tempos de serviço para todas as simulações foram gerados de acordo com uma distribuição exponencial com média 0,009. Como mostrado no capítulo 2, o parâmetro de Hurst mede o grau de auto-similaridade do processo e, além disso, vimos que quanto maior o parâmetro de Hurst em um processo auto-similar maior a dependência de longa duração (sua função de autocorrelação decai mais lentamente). Escolhemos os parâmetros de Hurst variando de 0,5 até 0,9 (mantendo sempre a mesma distribuição). As figuras abaixo mostram a distribuição de cauda do atraso desse sistema de cada uma das simulações. A distribuição de cauda indica a probabilidade do atraso no sistema ser maior que certo valor.

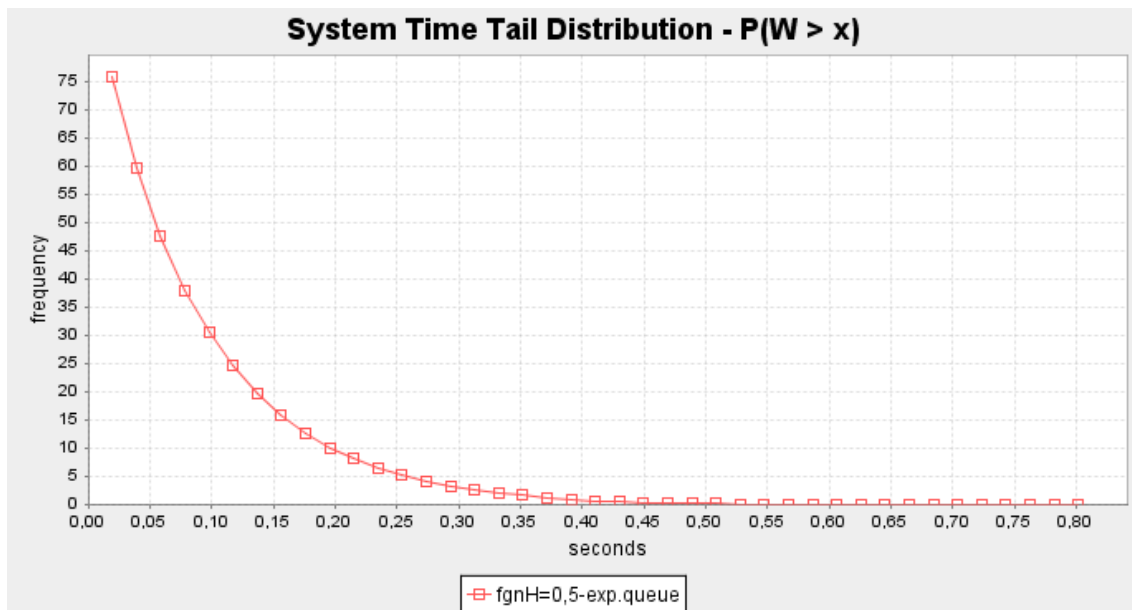


Figura 4.1 – Distribuição de cauda do atraso com o processo de chegada tendo parâmetro de Hurst igual a 0,5

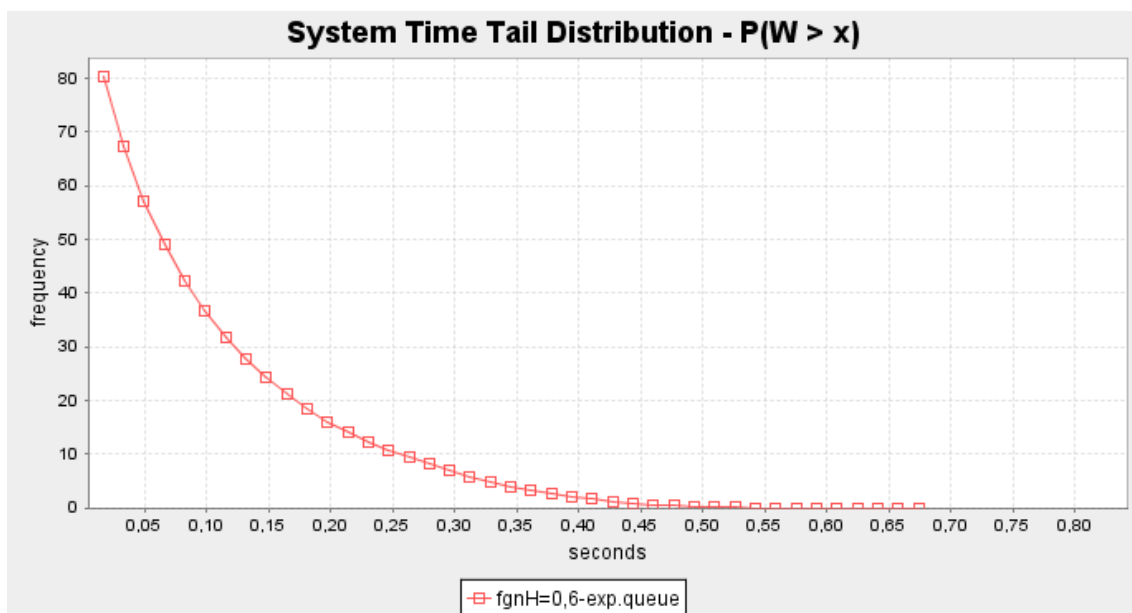


Figura 4.2 - Distribuição de cauda do atraso com o processo de chegada tendo parâmetro de Hurst igual a 0,6

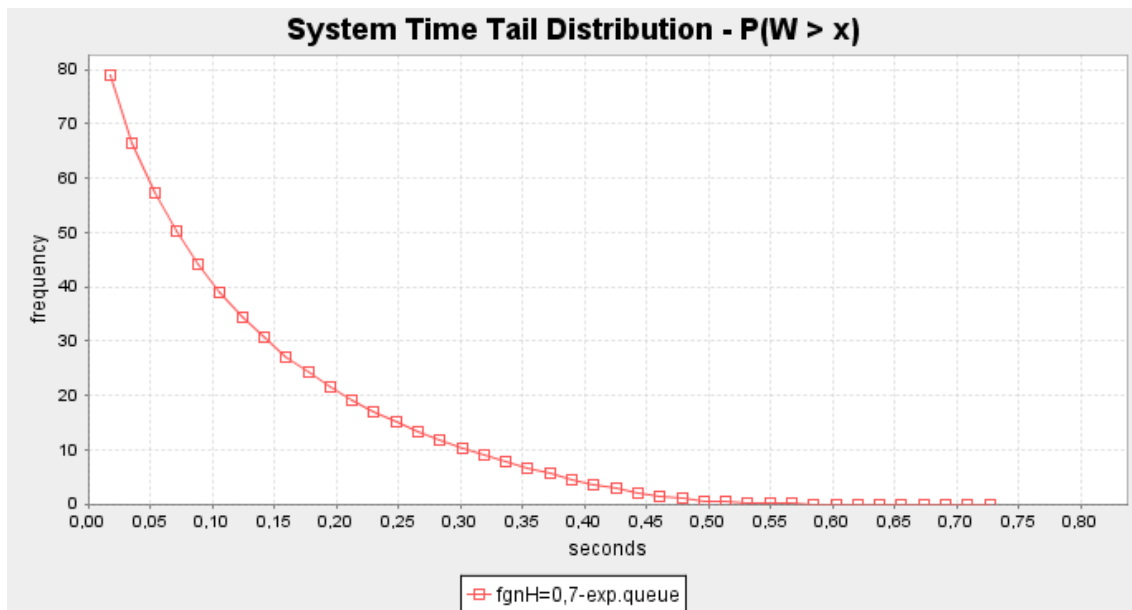


Figura 4.3 - Distribuição de cauda do atraso com o processo de chegada tendo parâmetro de Hurst igual a 0,7

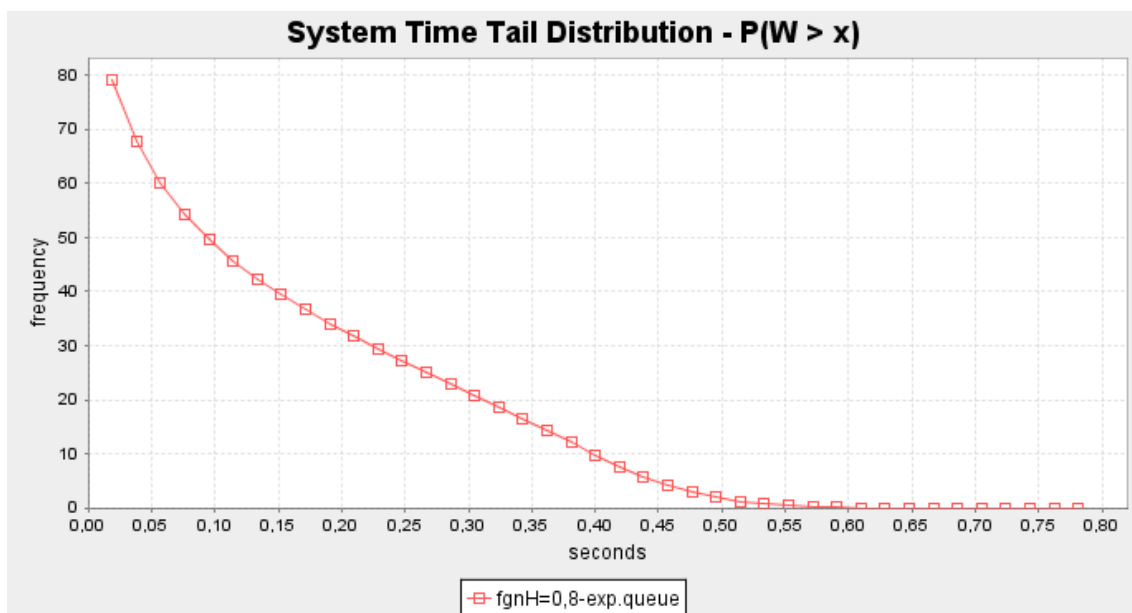


Figura 4.4 - Distribuição de cauda do atraso com o processo de chegada tendo parâmetro de Hurst igual a 0,8

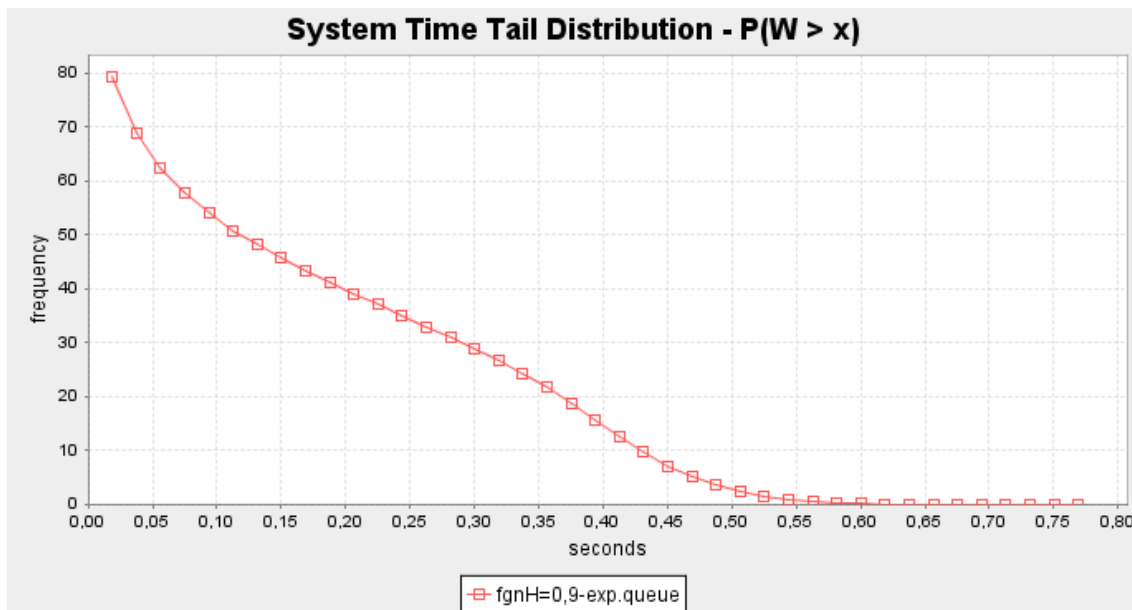


Figura 4.5 - Distribuição de cauda do atraso com o processo de chegada tendo parâmetro de Hurst igual a 0,9

Como pode ser visto pelas figuras acima, mesmo mantendo a mesma distribuição para os processos de chegada de todas as simulações (todos os processos de chegada possuem distribuição gaussiana com mesma média e variância), a distribuição de cauda do tempo de atraso no sistema é mais pesada quanto maior for o parâmetro de Hurst. Isso quer dizer que uma maior porcentagem dos pacotes possui atraso maior que determinado valor. Assim, percebemos que em alguns casos a modelagem do processo estocástico baseado somente em sua distribuição, desconsiderando qualquer dependência entre os valores de suas funções amostrais, pode ser enganosa e é necessário inferir sobre outras características estatísticas do processo para sua correta modelagem.

Nesse capítulo mostraremos métodos de se estimar diversos parâmetros de um processo estocástico, como média, variância e parâmetro de Hurst. Além disso, mostraremos métodos de se verificar a semelhança entre a distribuição do processo estocástico analisado e distribuições teóricas existentes.

4.1. Estimação de Média e Variância

Supondo X_1, X_2, \dots, X_n variáveis aleatórias independentes identicamente distribuídas com média finita μ e variância finita σ^2 . Nesse caso o estimador dado pela média amostral definido por (4.1) é um estimador de μ sem viés.

$$\hat{\mu}(n) = \frac{\sum_{i=1}^n X_i}{n} \quad (4.1)$$

Um estimador sem viés significa que, se realizarmos vários experimentos independentes, cada um com média amostral $\hat{\mu}(n)$, então a esperança dos $\hat{\mu}(n)$'s será μ . Isso pode ser demonstrado da seguinte forma:

$$E[\hat{\mu}(n)] = E\left[\frac{1}{n} \sum_{i=1}^n X_i\right] = \frac{1}{n} \sum_{i=1}^n E[X_i] = \frac{1}{n} n \cdot \mu = \mu$$

Pode se mostrar também que, para observações independentes, a variância amostral definida por (4.2) também é um estimador sem viés para σ^2 .

$$\hat{\sigma}^2(n) = \frac{\sum_{i=1}^n [X_i - \hat{\mu}(n)]^2}{n-1} \quad (4.2)$$

Demonstramos que esse estimador não possui viés da seguinte forma:

$$\begin{aligned} E[\hat{\sigma}^2(n)] &= E\left[\frac{1}{n-1} \sum_{i=1}^n [X_i - \hat{\mu}(n)]^2\right] = \frac{1}{n-1} E\left[\sum_{i=1}^n [X_i^2 - 2X_i\hat{\mu}(n) + \hat{\mu}(n)^2]\right] \\ (n-1)E[\hat{\sigma}^2(n)] &= \sum_{i=1}^n E[X_i^2] - 2E\left[\sum_{i=1}^n X_i\hat{\mu}(n)\right] + \sum_{i=1}^n E[\hat{\mu}(n)^2] \end{aligned}$$

Como $\sum_{i=1}^n X_i = n\hat{\mu}(n)$ então:

$$\begin{aligned} (n-1)E[\hat{\sigma}^2(n)] &= nE[X_i^2] - 2nE[\hat{\mu}(n)^2] + nE[\hat{\mu}(n)^2] = nE[X_i^2] - nE[\hat{\mu}(n)^2] \\ \frac{n-1}{n} &= E[X_i^2] - E[\hat{\mu}(n)^2] \end{aligned}$$

O valor de $E[\hat{\mu}(n)^2]$ é dado por:

$$E[\hat{\mu}(n)^2] = \text{var}[\hat{\mu}(n)] + (E[\hat{\mu}(n)])^2 = \text{var}\left[\frac{1}{n} \sum_{i=1}^n X_i\right] + \mu^2 = \frac{1}{n^2} \text{var}\left[\sum_{i=1}^n X_i\right] + \mu^2$$

$$E[\hat{\mu}(n)^2] = \frac{1}{n^2} \sum_{i=1}^n \sigma^2 + \mu^2 = \frac{1}{n} \sigma^2 + \mu^2$$

Logo temos:

$$\frac{n-1}{n} E[\hat{\sigma}^2(n)] = E[X_i^2] - \left(\frac{1}{n} \sigma^2 + \mu^2\right) = \sigma^2 + \mu^2 - \frac{1}{n} \sigma^2 - \mu^2 = \left(1 - \frac{1}{n}\right) \sigma^2 = \frac{n-1}{n} \sigma^2$$

$$E[\hat{\sigma}^2(n)] = \sigma^2$$

4.2. Especificação de Distribuições Teóricas Baseada em Dados Observados

A especificação da distribuição teórica que melhor aproxima um determinado processo estocástico envolve basicamente os três seguintes passos:

- Definição das famílias de distribuições
- Estimação dos parâmetros das distribuições
- Determinar o quão representativo é a distribuição escolhida

O primeiro passo consiste entre as famílias de distribuição (e.g. exponencial, gaussiana, weibull) que parecem ser apropriadas, sem se preocupar com os valores dos parâmetros dessas famílias de distribuição. Em alguns casos, um conhecimento teórico sobre o processo estocástico que se deseja modelar pode ajudar a selecionar as possíveis distribuições que podem modelar esse processo, mesmo antes de obter dados em relação ao processo. No entanto, dificilmente temos conhecimento teórico suficiente sobre o processo físico que nos permita selecionar apenas uma possível distribuição. Além desse conhecimento teórico, algumas medidas como média e variância pode nos dar uma idéia das possíveis distribuições. Por exemplo, no caso de uma distribuição exponencial, o coeficiente de variação $cv = \sigma/\mu$ é igual a 1, logo, um coeficiente de

variação amostral próximo de 1 sugere que a distribuição do processo seja exponencial. Outro método de se verificar as possíveis famílias de distribuição é verificando a forma do histograma, que é basicamente uma estimativa gráfica da função densidade de probabilidade.

Depois de definir um ou mais candidatos para as distribuições, deve-se especificar os valores de seus parâmetros para especificar completamente as possíveis distribuições a serem usadas. Dois métodos clássicos para estimação dos parâmetros são o método da máxima semelhança (MLE) e o método dos momentos (MOM). O método dos momentos estima o parâmetro igualando os momentos teóricos aos momentos amostrais (caso o parâmetro possa ser dado em função dos momentos da distribuição) e resolvendo a equação em função do parâmetro. Por exemplo, no caso de uma distribuição exponencial, o parâmetro λ é igual a $1/\mu$, em que μ é a média do processo. Pelo método dos momentos, o estimador de λ é:

$$\hat{\lambda} = \frac{1}{\hat{\mu}} = \frac{n}{\sum_{i=1}^n X_i}$$

O estimador de máxima semelhança é definido da seguinte forma. Primeiro definimos a função de semelhança em função do parâmetro θ desejado como sendo a função densidade probabilidade conjunta de se obter os n valores observados para aquela distribuição. Ou seja, caso os valores observados X_1, X_2, \dots, X_n sejam i.i.d. a função de máxima semelhança é dada por:

$$L(\theta) = p_{\theta}(X_1)p_{\theta}(X_2)\dots p_{\theta}(X_n)$$

Assim, o estimador de máxima semelhança é definido como sendo o valor de θ que maximiza a função de máxima semelhança. A idéia por trás desse estimador é mais facilmente percebida para o caso de uma distribuição discreta. Nesse caso, a função de máxima semelhança dá a probabilidade de se obter os n valores observados para a distribuição escolhido com dado parâmetro θ . Assim o parâmetro que maximize $L(\theta)$ é o parâmetro que melhor explica os dados observados para determinada distribuição. Para o caso de uma distribuição exponencial, o MLE é obtido da seguinte forma.

Primeiro definimos a função de máxima semelhança como sendo (para o caso de variáveis independentes identicamente distribuídas):

$$L(\lambda) = \prod_{i=1}^n p_{\lambda}(X_i) = \prod_{i=1}^n \lambda e^{-\lambda X_i} = \lambda^n e^{-\lambda \sum_{i=1}^n X_i}$$

Nesse caso, é mais fácil acharmos λ que maximiza o logaritmo natural de $L(\lambda)$, que é o mesmo λ que maximiza $L(\lambda)$. Assim obtemos:

$$\max_{\lambda} [\ln(L(\lambda))] = \max_{\lambda} [n \ln(\lambda) - \lambda \sum_{i=1}^n X_i]$$

Calculando a derivada de $\ln(L(\lambda))$ em relação a λ e igualando a zero, obtemos o valor máximo de λ .

$$\frac{\partial}{\partial \lambda} \ln(L(\lambda)) = \frac{n}{\lambda} - \sum_{i=1}^n X_i = 0$$

Logo o MLE da distribuição exponencial é dado por:

$$\hat{\lambda} = \frac{n}{\sum_{i=1}^n X_i}$$

Podemos ver que para o caso da distribuição exponencial o estimador MLE e o estimador MOM são equivalentes. No entanto, esse não é o caso para todas as distribuições. O estimador MLE costuma ter propriedades estatísticas melhores que o método dos momentos e deve ser escolhido quando possível.

Após determinar completamente uma ou mais distribuições que podem corresponder aos dados observados, seguindo os dois primeiros passos explicados anteriormente, devemos então examinar melhor essas distribuições para verificar o quão bem elas representam a distribuição do processo estocástico e determinar qual distribuição é a melhor escolha. Para isso, normalmente são utilizados os chamados *goodness-of-fit tests*. Esses testes são testes de hipóteses estatísticas usados para definir se as observações X_1, X_2, \dots, X_n são amostras independentes de uma determinada distribuição F.

Um dos mais usados e mais antigos teste de hipótese é o teste do qui-quadrado. O teste do qui-quadrado pode ser pensado como uma comparação formal de um histograma com a função densidade de probabilidade teórica. Para calcular o valor do qui-quadrado, primeiro deve-se dividir a distribuição em k intervalos adjacentes. Em seguida se verifica o número N_j de observações em cada intervalo. Em seguida calculamos a proporção esperada p_j de ocorrências em cada intervalo para a distribuição desejada. Para o caso contínuo temos:

N_j = número de X_i 's no j -ésimo intervalo $[a_{j-1}, a_j)$

$$p_j = \int_{a_{j-1}}^{a_j} f(x) dx$$

sendo $f(x)$ a função densidade probabilidade da distribuição. Assim, o valor do qui-quadrado é dado por:

$$\chi^2 = \sum_{j=1}^k \frac{(N_j - np_j)^2}{np_j} \quad (4.3)$$

Como np_j é a quantidade esperada de observações que cairiam no j -ésimo intervalo, se a hipótese fosse verdadeira, o valor de χ^2 deveria ser pequeno. Logo, rejeitaríamos a hipótese caso χ^2 fosse muito grande.

Pode ser provado que, se todos os parâmetros da distribuição forem conhecidos de antemão e se os valores observados realmente corresponderem a variáveis aleatórias independentes com a distribuição teórica estimada, então o valor obtido pelo χ^2 converge, à medida que $n \rightarrow \infty$, para uma distribuição de χ^2 com $k-1$ graus de liberdade, em que k é o número de intervalos. Então podemos definir um valor limite $\chi^2_{k-1, 1-\alpha}$ para o qual aceitaríamos a hipótese. O valor limite corresponde ao ponto em que, para uma variável aleatória X que possua uma distribuição de qui-quadrado com $k-1$ graus de liberdade, a probabilidade de X ser maior que esse valor seja $P(X > \chi^2_{k-1, 1-\alpha}) = \alpha$. Ou seja, a probabilidade de errarmos rejeitando a hipótese seria de α .

No caso de termos m parâmetros estimados a partir dos dados, então a distribuição dos χ^2 's calculados caso a hipótese seja verdadeira não seguirá uma

distribuição de qui-quadrado com grau de liberdade $k-1$. No entanto, caso o método utilizado seja o MLE, pode ser provado que a distribuição dos χ^2 's calculados estão entre as funções de distribuição de qui-quadrado com $k-m-1$ graus de liberdade e com $k-1$ graus de liberdade. Nesse caso, fica claro que devemos aceitar a hipótese caso o valor calculado seja menor que $\chi^2_{k-m-1,1-\alpha}$ e rejeitar caso ele seja maior que $\chi^2_{k-1,1-\alpha}$. No entanto não fica claro o que deve ser feito caso o valor calculado esteja entre $\chi^2_{k-m-1,1-\alpha}$ e $\chi^2_{k-1,1-\alpha}$. Conservativamente é aconselhado rejeitar a hipótese somente caso o valor calculado seja maior que $\chi^2_{k-1,1-\alpha}$, pois assim garantimos que a probabilidade de errar rejeitando-se a hipótese é menor ou igual a α . No entanto isso diminui o poder (probabilidade de se rejeitar uma hipótese falsa) do teste. Caso m não seja maior que 2, e caso o número de graus de liberdade seja suficientemente grande, a diferença entre os valores $\chi^2_{k-m-1,1-\alpha}$ e $\chi^2_{k-1,1-\alpha}$ não será muito grande e a perda do poder do teste ao seguir a recomendação mais conservativa não será muito grande.

Uma das maiores dificuldades de se implementar um teste de qui-quadrado consiste em escolher o tamanho e o número de intervalos. Existem basicamente dois modos de se escolher os intervalos: intervalos iguais ou probabilidades iguais. No primeiro caso determinamos o maior e o menor valor da amostra e dividimos em k intervalos iguais. No segundo caso, a partir da distribuição que está sendo avaliada, escolhemos os k intervalos de forma que todos os p_j 's tenham o mesmo valor. Em qualquer um dos casos o número de intervalos deve ser escolhido de forma que cada intervalo possua um número significativo de amostras. Geralmente é aconselhado que o método escolhido seja o método de probabilidades iguais de forma que $np_j \geq 5$ para todo j (no caso contínuo).

4.3. *Estimação do Parâmetro de Hurst*

Estimar o parâmetro de Hurst é algo muito difícil mesmo para traços gerados artificialmente, como pode ser verificado pelos testes de [12] e [13]. Na análise de traços reais, essa estimação se torna ainda mais difícil. Um processo que não possua dependência de longa duração e apresenta componentes com certa periodicidade pode enganar os estimadores, podendo revelar, falsamente, valores altos para o parâmetro de Hurst. Além disso, *trends* presentes nos dados podem também levar a conclusões

erradas. Para traços reais, muitas vezes é necessário realizar um pré-processamento dos dados, de forma a tentar remover qualquer periodicidade percebida ou qualquer “trend” presente no traço.

Devido a toda a dificuldade na estimação do parâmetro de Hurst, nunca é aconselhável confiar em apenas um tipo de estimador e deve ser sempre fornecido, junto com a estimativa, o método utilizado. Apresentaremos nessa seção quatro dos mais usados métodos:

- Estimador Whittle
- Método Variance-Time Plot
- Método de Kettani-Gubner
- Método do Periodograma

4.3.1. Estimador Whittle

Suponha que $\{x_t\}$ represente a função amostra de um processo auto-similar X que possua parâmetro de Hurst desconhecido. Para estimar o parâmetro de Hurst, primeiro determinamos a densidade espectral de potência $f(\lambda, H)$ do processo X normalizado para ter variância 1, e o periodograma $I(\lambda)$ da função amostra $\{x_t\}$. Como o processo X é um processo auto-similar, sua densidade espectral de potência pode ser escrita em função do parâmetro de Hurst. Então para estimarmos o parâmetro de Hurst, achamos o valor de \hat{H} que minimiza a seguinte integral:

$$g(\hat{H}) = \int \frac{I(\lambda)}{f(\lambda; \hat{H})} \quad (4.4)$$

Esse método se baseia no fato que, caso X seja um processo auto-similar, as ordenadas do periodograma de sua função amostral são assintoticamente independentes e exponencialmente distribuídas com média $f(\lambda, H)$. Caso o processo analisado seja realmente auto-similar, esse método apresenta boas estimações do parâmetro de Hurst. Duas desvantagens desse método é que ele é mais custoso computacionalmente que os

demais métodos aqui apresentados e que, como devemos calcular a densidade espectral de potência do processo, ele é específico para cada tipo de processo auto-similar. Ou seja, o estimador para um processo auto-similar fGn deve ser diferente que o estimador para um outro processo auto-similar (e.g. fARIMA), já que suas densidades espectrais de potência são diferentes.

4.3.2. Método Variance-Time Plot

O método variance-time plot é baseado na propriedade indicada por (2.18) demonstrada anteriormente para processos auto-similares. De acordo com essa propriedade temos que:

$$\text{var}(X^{(m)}) = c_1 m^{-\beta}, \text{ com } H = 1 - \frac{\beta}{2}$$

Tirando o logaritmo dessa equação temos que:

$$\log[\text{var}(X^{(m)})] = \log(c_1) - \beta \log(m) \quad (4.5)$$

Para se estimar o parâmetro de Hurst, primeiro plotamos o gráfico $\log(\text{var}(X^{(m)}))$ vs $\log(m)$ e calculamos a inclinação β da reta que melhor aproxima esse gráfico. O parâmetro de Hurst então é dado por $H = 1 - \beta/2$.

4.3.3. Método de Kettani-Gubner

Esse método foi apresentado por Houssain Kettani e John A. Gubner em [11] e se baseia no coeficiente de autocorrelação de um processo auto-similar dado por (2.16). Como visto anteriormente um processo é exatamente auto-similar de segunda ordem quando seu coeficiente de autocorrelação é descrito pela seguinte equação:

$$\rho(k) = \frac{1}{2} \left[|k+1|^{2H} + |k-1|^{2H} - 2|k|^{2H} \right]$$

Para $k = 1$ temos:

$$\rho(1) = 2^{2H-1} - 1$$

É definido então um estimador para o parâmetro de Hurst como sendo:

$$\hat{H} = \frac{1}{2} [1 + \log(1 + \hat{\rho}_n(1))] \quad (4.6)$$

Em que $\hat{\rho}_n(k)$ é o coeficiente de autocorrelação amostral dado por:

$$\hat{\rho}_n(k) = \frac{\hat{C}_n(k)}{\hat{C}_n(0)}$$

$$\hat{C}_n(k) = \frac{1}{n} \sum_{i=1}^{n-k} (X_i - \hat{\mu}(n))(X_{i+k} - \hat{\mu}(n))$$

Onde $\hat{\mu}(n)$ é a média amostral definida anteriormente.

4.3.4. Método do Periodograma

Esse método baseia-se na forma da densidade espectral de potência de um processo com dependência de longa duração, dado pela equação (2.12). Como vimos, a densidade espectral de potência perto da origem possui a seguinte forma:

$$f_X(\lambda) \sim c_2 |\lambda|^{-\gamma}, \lambda \rightarrow 0, \gamma \in (0,1)$$

Tirando o logaritmo da equação acima obtemos:

$$\log(f_X(\lambda)) = \log(c_2) - \gamma \log(\lambda)$$

Comparando as equações (2.11) e (2.13), percebemos que, para o caso de um processo pelo menos assintoticamente auto-similar de segunda ordem, $\gamma = 2H - 1$. Como o periodograma $I(\lambda)$ é uma estimativa da densidade espectral de potência do processo, o parâmetro de Hurst pode ser estimado a partir do gráfico do periodograma. Para isso basta plotarmos o gráfico $\log(I(\lambda))$ vs $\log(\lambda)$ para valores pequenos de λ e estimarmos γ a partir da inclinação da reta que melhor aproxima esse gráfico. O parâmetro de Hurst é então estimado por:

$$\hat{H} = \frac{\gamma + 1}{2} \quad (4.7)$$

5. Geração de Tráfego

Para simularmos um tráfego na rede devemos determinar os tempos de saída (ou chegada) de cada pacote. Como um tráfego real é um processo estocástico, esses valores são aleatórios. Outra possível fonte de aleatoriedade de um tráfego é o tamanho de seus pacotes. Então, para simularmos um tráfego utilizamos dois geradores de variáveis aleatórias: um para gerarmos os tempos entre dois pacotes sucessivos (determinando os tempos de saída ou chegada de cada pacote) e outro para gerarmos os tamanhos de cada pacote. Nesse capítulo, mostraremos alguns métodos para geração de variáveis de um determinado processo aleatório.

Como veremos a seguir, uma ferramenta básica para a geração de variáveis aleatórias seguindo qualquer distribuição é uma fonte de variáveis aleatórias independentes e identicamente distribuídas com distribuição uniforme entre 0 e 1. A maior parte das linguagens de programação possuem geradores de números pseudo-aleatórios. Eles são “pseudo” por serem totalmente reproduzíveis por um algoritmo matemático, mas são “aleatórios” porque passam no testes estatísticos, que testam basicamente se os valores são independentes e possuem distribuição uniforme. Grande parte dos geradores pseudo-aleatórios presentes nas linguagens de programação é baseada no método congruo-linear. Esse método é basicamente um algoritmo recursivo da forma:

$$r_{n+1} = (r_n k + a) \bmod m \quad (5.1)$$

Em que k , a e m são números inteiros positivos, com $k < m$ e $a < m$. Um valor inicial r_0 , chamado semente, deve ser escolhido para ser o primeiro elemento da sequência de números pseudo-aleatórios. Esse algoritmo gera uma sequência periódica com período máximo de $m-1$. Os valores escolhidos para a , k e m determinarão se a sequência gerada terá características estatísticas satisfatórias, ou seja, se seus valores não podem ser distinguidos de variáveis aleatórias i.i.d com distribuição uniforme. Um grande número de valores para a , k e m já foi testado e existem na literatura diversos valores que determinam sequências aleatórias satisfatórias. De acordo com [9], valores de m muito usados em softwares de simulação são $m = 2^{31}-1$ e $m = 2^{48}$.

5.1. Geração por Inversão

Esse método pode ser usado para se gerar variáveis aleatórias i.i.d. que possuam uma distribuição conhecida, ou mesmo de uma distribuição empírica, contanto que seja definido a c.d.f. $F(x)$ da distribuição. O algoritmo é muito simples e consiste basicamente de dois passos. Primeiro gera-se uma variável aleatória U com distribuição uniforme entre 0 e 1. A variável aleatória desejada é dada por $X = F^{-1}(U)$, em que $F^{-1}(x)$ é o inverso da c.d.f. desejada. Esse método é ilustrado pela figura 5.1.

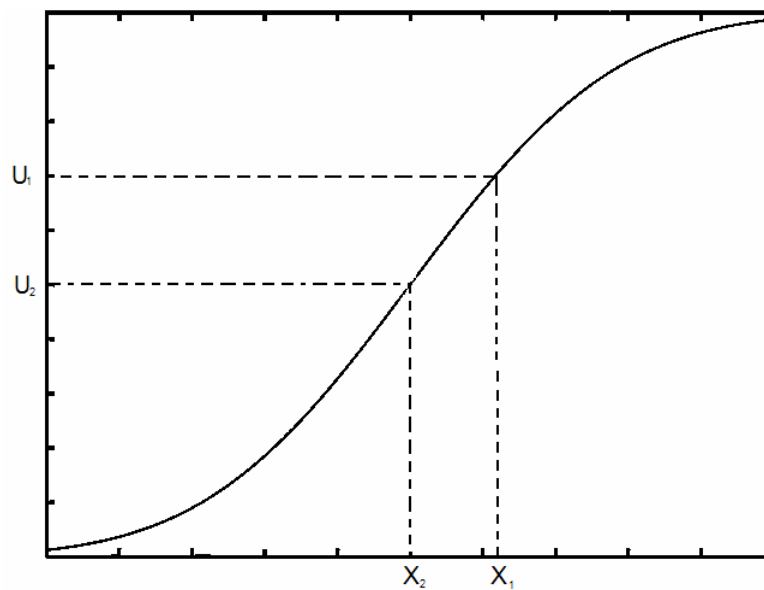


Figura 5.1 – Método de Geração por Inversão

É facilmente demonstrável que a variável aleatória X gerada por esse método possui a distribuição desejada. Como $F^{-1}(U) = X$, temos que: $P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x))$. Como U é uniformemente distribuído entre 0 e 1, então $P(U \leq a) = a$. Logo, $P(X \leq x) = F(x)$ e a variável aleatória X possui a c.d.f. desejada.

5.2. Geração de Processo fGn

Um processo fGn não pode ser gerado pelo método da geração por inversão, pois nesse caso não basta gerarmos uma variável aleatória i.i.d. com distribuição gaussiana. Por ser um processo auto-similar, devemos modelar também o efeito do parâmetro de Hurst que influencia diretamente a função de autocorrelação do processo.

Ou seja, os valores gerados devem ser correlacionados de forma que a função amostra possua o parâmetro de Hurst desejado.

O método que escolhemos para a geração de um processo fGn se baseia no espectro de potência dos processos fGn e foi apresentado por Vern Paxson em [ref]. Esse método gera um processo auto-similar em tempo discreto $G(k)$ como o definido em (2.14), onde $G(k)$ representa o número de ocorrências no k -ésimo intervalo de tamanho δ . No caso da geração de tráfego cada ocorrência representa um pacote enviado (ou recebido), ou seja, $G(k)$ representa o número de pacotes enviados no k -ésimo intervalo.

A geração de uma função amostra fGn consiste nos seguintes passos:

1. Primeiro construímos uma seqüência de valores $\{f_1, f_2, \dots, f_{n/2}\}$ correspondentes ao espectro de potência de um processo fGn para as frequências de $2\pi/n$ a π
2. Em seguida multiplicamos cada f_i para variáveis aleatórias exponenciais i.i.d com média 1, obtendo a seqüência $\{\hat{f}_i\}$. Isso é feito porque, caso X seja um processo auto-similar, as ordenadas do periodograma de sua função amostral são assintoticamente independentes e exponencialmente distribuídas com média $f(\lambda, H)$, em que $f(\lambda, H)$ é o valor do espectro de potência de um processo fGn com parâmetro H na frequência λ . Dessa forma temos que, caso seja grande o suficiente, $\{\hat{f}_i\}$ representa o periodograma de uma função amostra fGn.
3. Construímos então uma seqüência de valores complexos $\{z_1, \dots, z_{n/2}\}$ tal que $|z_i| = \sqrt{\hat{f}_i}$ e a fase de z_i seja uniformemente distribuída entre 0 e 2π . Essa fase aleatória preserva o espectro de potência do processo, já que o espectro de potência depende unicamente do módulo de z_i , mas garante que diferentes funções amostras geradas por esse método sejam diferentes. Além disso, foi verificado que essa fase aleatória também faz com que as distribuições marginais do resultado final sejam gaussianas,

um requerimento para o processo fGn. A sequência $\{z_i\}$ representa a transformada de Fourier discreta do sinal para as frequências positivas.

4. Expandimos a sequência $\{z_i\}$, obtendo a sequência $\{z'_i\}$, que representa a transformada de Fourier discreta para frequências positivas e negativas. Como desejamos um sinal real, devemos expandir $\{z_i\}$ de forma que seu módulo seja uma função par e sua fase seja uma função ímpar. A sequência $\{z'_i\}$ é construída da seguinte forma:

$$z'_j = \begin{cases} 0 & \text{se } j = 0 \\ z_j & \text{se } 0 < j < n/2 \\ \bar{z}_{n-j} & \text{se } n/2 < j < n \end{cases}$$

Em que \bar{z}_{n-j} representa o conjugado complexo de z_{n-j} .

5. Por fim, calculamos a transformada de Fourier inversa de $\{z'_i\}$ para obtermos a função amostra fGn $\{x_i\}$.

Como esse método é criado a partir do espectro de potência de um processo fGn, ele possui propriedades de autocorrelação de um processo fGn, já que a autocorrelação de um processo estocástico e seu espectro de potência formam um par de Transformada de Fourier. A sequência gerada por esse método possui média 0. No entanto, como em um processo fGn a média, variância e parâmetro de Hurst são independentes, podemos obter qualquer média e variância desejadas por meio de transformações lineares, sem alterar o parâmetro de Hurst.

Como já foi dito anteriormente, no caso da geração de um tráfego, cada valor da sequência gerada por esse método representa o número de pacotes em um intervalo de tamanho δ . Para gerarmos o tráfego devemos obter os tempos de saída (ou chegada) de cada pacote e não a quantidade de pacotes por intervalo de tempo. Para isso, devemos definir o tamanho de cada intervalo δ e distribuímos os pacotes dentro desse intervalo. Como só podemos enviar um número inteiro de pacotes em um intervalo δ , devemos converter os valores reais da sequência obtida para valores inteiros. O problema maior então está em distribuímos os valores dentro do intervalo de forma a garantirmos a

auto-similaridade para escalas menores que δ . No entanto, como vários estudos indicam que os tráfegos reais são apenas assintoticamente auto-similares, ou seja, não são auto-similares para escalas pequenas, distribuímos os pacotes de maneira uniforme dentro do intervalo δ , garantindo a auto-similaridade apenas para escalas maiores que essa, e definimos δ pequeno.

5.3. *WIG – Wavelet-domain independent Gaussian model*

Esse método é capaz de gerar um processo gaussiano capturando suas características de autocorrelação. Ele se baseia no fato de que a transformada de wavelet descorrelaciona sinais que possuam dependência de longa duração. Dessa forma, o difícil problema de se modelar um processo altamente correlacionado no domínio do tempo se reduz a um simples problema de modelá-lo por um processo praticamente descorrelacionado no domínio das wavelets. A família de wavelets usada nesse método é a wavelet de Haar.

Esse método consiste em gerarmos os coeficientes de escala U_{0k} da correspondente a escala mais ampla do processo a partir de uma variável aleatória Gaussiana e gerarmos os coeficientes de wavelet W_{jk} como variáveis aleatórias Gaussianas independentes (não-correlacionadas) e identicamente distribuídas com média 0 de acordo com $W_{jk} \sim N(0, \sigma_j^2)$, sendo σ_j^2 a variância dos coeficientes de wavelet na escala j . Dessa forma, a partir de (D.18) e (D.19), podemos chegar na representação em tempo discreto do processo na escala n desejada gerando os coeficientes de wavelet W_{jk} para as escalas de 0 até $n - 1$.

Os σ_j^2 's para as diferentes escalas influencia o comportamento da autocorrelação do processo. Para o caso de um processo fGn, pode ser provado que os σ_j^2 's são dados por (2) (ver [20]).

$$\sigma_j^2 = \sigma^2 2^{(2H-1)(j-1)} (2 - 2^{2H-1}) \quad (5.2)$$

Na geração do processo, os σ_j^2 's podem ser definidos a partir de uma lei como (5.2). Nesse caso seria gerarmos um processo $G(k)$ fGn como o processo gerado pelo

método do espectro de potência explicado anteriormente. Podemos estabelecer os σ_j^2 's também a partir de dados reais, verificando as variâncias amostrais dos coeficientes de wavelet do processo que se deseja simular. Esse método de estabelecer os σ_j^2 's possui a vantagem de que o traço simulado apresentará as dependências de longa e curta duração mais similares às dos valores reais obtidos. A desvantagem desse método é que ele é muito menos compacto de se modelar o tráfego. No caso de estabelecermos os σ_j^2 's a partir de uma lei, poucos parâmetros, como o parâmetro de Hurst definem o processo, já no caso de estabelecermos os σ_j^2 's a partir de dados reais devemos ser capazes de obter um grande número de dados para definir o processo, o que em alguns casos pode ser muito difícil.

Assim como no caso da geração de tráfego fGn pelo espectro de potência, o processo obtido por esse método define os número de pacotes por intervalo e, assim como no caso anterior, devemos distribuir esses pacotes dentro do intervalo a fim de gerarmos o tráfego.

5.4. *MWM – Multifractal Wavelet Model*

Esse modelo, apresentado em [20], se baseia no mesmo princípio do modelo WIG, que modela o processo altamente correlacionado no tempo como um processo decorrelacionado no domínio de wavelets. No entanto, ao invés de gerar um processo com distribuição gaussiana, que pode apresentar quaisquer valores entre $-\infty$ e $+\infty$, os coeficientes são gerados de forma a garantir que os valores do sinal serão todos positivos. A distribuição do processo obtido a partir desse modelo apresenta uma cauda mais pesada que a distribuição normal, o que foi observado para alguns tipos de tráfegos reais, principalmente em escalas pequenas [19] Geralmente os tráfegos reais possuem distribuições normais para escalas pequenas apenas quando eles possuem um nível de agregação muito alto, possuindo uma distribuição de cauda mais pesada quando possuem um nível de agregação médio.

Como as wavelets utilizadas são as wavelets de Haar, os coeficientes de escala U_{jk} representam o próprio sinal em tempo discreto na escala j . Assim, o sinal será não-negativo se e somente se os coeficientes U_{jk} 's forem positivos para qualquer j e k . A

partir de (D.18) e (D.19), podemos perceber que garantimos a não-negatividade dos coeficientes $U_{j+1,k}$'s se os coeficientes U_{jk} 's forem positivos e se $|W_{jk}| \leq U_{jk}$, $\forall j,k$.

A partir dessas condições chegamos a um simples algoritmo para geração do tráfego. Primeiro geramos os coeficientes U_{0k} a partir de uma distribuição gaussiana. Na verdade, para garantirmos que o processo seja estritamente positivo, os U_{0k} deveriam ser gerados a partir de uma distribuição estritamente positiva. No entanto, se escolhermos a escala '0' ampla o suficiente de forma a garantirmos que a média do processo nessa escala seja muito maior que a sua variância, esses valores podem ser escolhidos a partir de uma gaussiana. Para chegarmos aos coeficientes na escala n desejada, devemos então gerar os coeficientes de wavelet W_{jk} de forma a garantir que $|W_{jk}| \leq U_{jk}$, $\forall j,k$. Então, os coeficientes W_{jk} são calculados recursivamente a partir de (5.3).

$$W_{jk} = A_{jk} U_{jk} \quad (5.3)$$

Em que os A_{jk} 's são variáveis aleatórias entre $[-1,1]$. Logo, a partir de U_{0k} , podemos calcular a projeção do sinal em escalas mais finas recursivamente pelas equações (5.4) e (5.5).

$$U_{j+1,2k} = \frac{1}{\sqrt{2}}(1 + A_{jk})U_{jk} \quad (5.4)$$

$$U_{j+1,2k+1} = \frac{1}{\sqrt{2}}(1 - A_{jk})U_{jk} \quad (5.5)$$

Como foi dito, a distribuição dos A_{jk} escolhidos deve ser restrita ao intervalo $[-1,1]$. Além disso, para que o processo seja estacionário de primeira ordem, eles devem ser simétricos com relação ao 0 e devem ser identicamente distribuídos dentro da mesma escala. A distribuição utilizada que possui todas essas características é a distribuição beta simétrica $\beta_{-1,+1}(p_j, p_j)$.

A distribuição beta está ilustrada na figura 5.2 para diferentes valores de p_j . Se p_j for igual a um, a distribuição beta corresponde a distribuição uniforme entre 0 e 1 e quanto maior for o p_j mais ele se aproxima de uma distribuição gaussiana truncada entre zero e um.

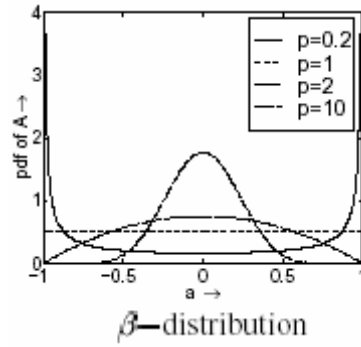


Figura 5.2 – Distribuição Beta

A autocorrelação do processo é controlada pelos p_j 's escolhidos, que possuem a seguinte relação com a variância dos coeficientes de wavelet na escala j σ_j^2 :

$$\frac{\sigma_{j-1}^2}{\sigma_j^2} = \frac{2p_j + 1}{p_{j-1} + 1} \quad (5.6)$$

Os σ_j^2 's podem ser escolhidos da mesma forma que pelo método WIG, ou seja, a partir de uma lei como (5.2) ou a partir de valores reais extraídos de um processo real.

6. GTAR

O software GTAR (Gerador de Tráfego e Analisador de Rede) apresentado em [7] é uma ferramenta de código aberto e que tem como objetivo principal a análise de uma rede de acordo com a seguinte estrutura:

1. Estabelecimento de sincronismo entre diversos elementos dessa rede com um intervalo de acurácia de 1ms.
2. Geração de tráfegos, entre os diversos elementos sincronizados, de acordo com um processo estocástico.
3. Análise dos resultados coletados:
 - a. Atraso sofrido por cada pacote entre origem e destino.
 - b. Variação do atraso entre pacotes sucessivos (jitter).
 - c. Vazão (em bps) de cada tráfego que é inserido na rede pela ferramenta.
 - d. Análise estatística do processo de saída e de chegada de pacotes na origem e no destino respectivamente.

Dessa forma a ferramenta está dividida em três módulos básicos:

1. Módulo de sincronismo.
2. Módulo de geração de tráfego.
3. Módulo de inferência e análise de performance.

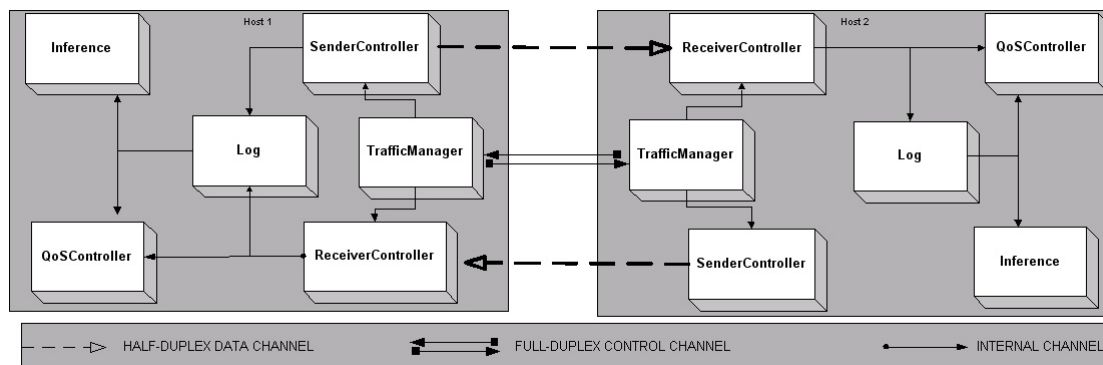



Figura 6.1 - Arquitetura do GTAR (extraída de [7])

Nessa figura mostramos a estrutura dos módulos de geração de tráfego e de inferência e análise de performance.

O módulo de geração de tráfego é composto pelo *TrafficManager* que estabelece como será o tráfego, ou seja, o tempo entre os pacotes e tamanho dos pacotes definidos de acordo com a distribuição selecionada, *SenderController* que recebe as definições do tráfego do *TrafficManager* e envia os pacotes, *ReceiverController* que recebe os pacotes de um tráfego pré-estabelecido e envia ao *Log* e ao *QoSController* as informações relevantes de cada pacote (tamanho, tempo de saída, tempo de chegada e número do pacote) e *Log* que escreve as informações recebidas em arquivos próprios.

O módulo de inferência é composto do *QoSController* que faz a análise das medidas de QoS a partir dos dados obtidos no *Log* ou diretamente com o *ReceiverController* e do *Inference* que busca estimar valores estatísticos do tráfego (média, variância, parâmetro de Hurst e adequação a certa distribuição matemática)

Complementando a apresentação da ferramenta feita em [7], apresentaremos algumas funções e o modo de inserção de dados para inferência.

Na tela principal da ferramenta do GTAR ao clicar no submenu ‘Abrir’ do menu ‘Arquivo’ ou simplesmente no botão  na barra de ferramentas, será aberta a janela ilustrada na figura 6.2.

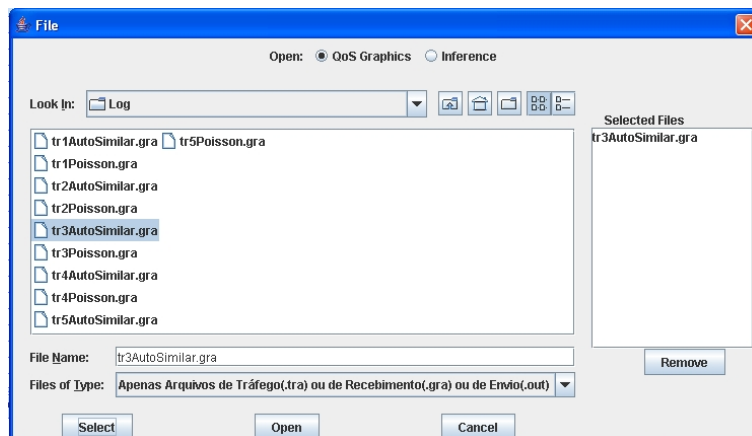


Figura 6.2 – Janela: ‘Abrir arquivo’

Para abrir o arquivo que contém os dados que se deseja realizar a inferência, devem ser realizados os seguintes procedimentos: selecionar ‘*Inference*’ no alto da janela, selecionar o arquivo desejado, clicar em ‘*Select*’, o arquivo selecionado irá aparecer na janela ‘*Selected Files*’, se houver algum arquivo não desejado nessa janela este deverá ser selecionado e removido, e finalmente clicar em ‘*Open*’. A janela que aparecerá em seguida será equivalente à exibida na figura 6.3.

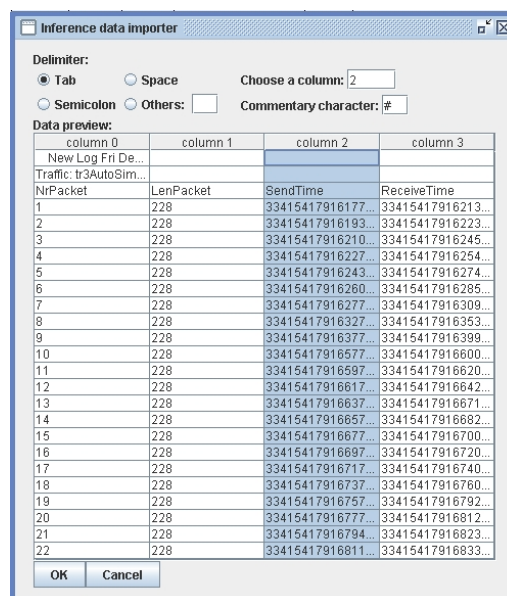


Figura 6.3 – Janela: ‘Importador de dados externos’ – Selecionar coluna

Nessa janela é possível a separação dos dados contidos no arquivo em colunas e a inferência será realizada em relação a apenas uma coluna selecionada. Assim é possível importar arquivos de texto de qualquer ferramenta que gere arquivos de *log*. A separação das colunas nos arquivos de texto pode ser feita com qualquer caractere. Após

a seleção da coluna desejada deve-se clicar no botão ‘OK’. A janela que aparecerá em seguida será equivalente à exibida na figura 6.4.

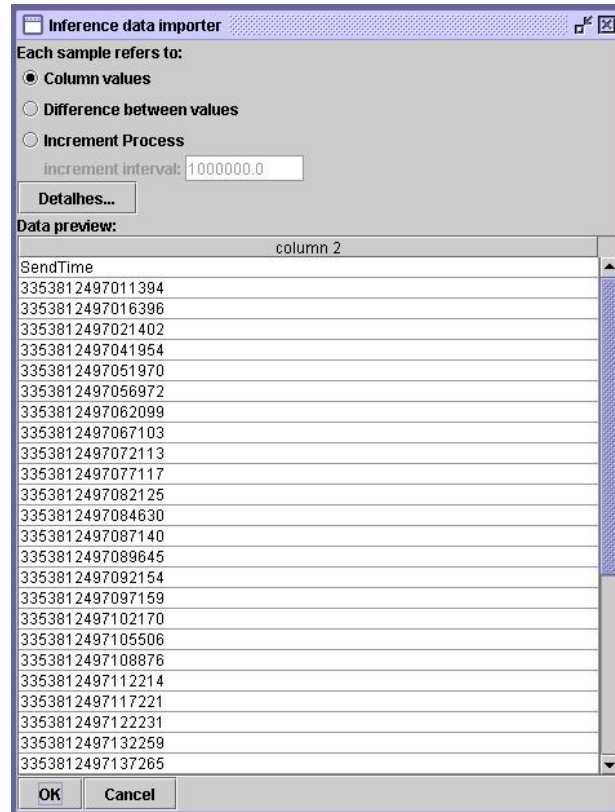


Figura 6.4 – Janela: ‘Importador de dados externos’ – Selecionar modo de análise dos dados

Nessa janela é possível definir como será a análise dos dados contidos na coluna selecionada na janela anterior. São definidos três tipos de análise distintos: valor literal, diferença entre valores consecutivos e processo de incremento. Na análise da diferença entre os valores, caso os dados representem os tempos de chegada dos pacotes a análise será sobre os tempos entre chegadas. Na análise do processo de incremento, sendo o mesmo caso anterior, a análise será sobre a quantidade de chegadas que ocorreram em intervalos sucessivos de tamanho definido pelo campo intervalo do incremento. Para alterar o valor do intervalo do incremento e visualizar a alteração no *preview*, é necessário pressionar a tecla ‘Enter’ após alterar o valor. Após a seleção do modo de análise desejada deve-se clicar no botão ‘OK’. Na janela que aparecerá em seguida poderão ser exibidos diversos gráficos e será possível o cálculo de vários parâmetros do processo. A janela que aparecerá em seguida e após selecionar o submenu ‘Distribution’ do menu ‘Graphics’ será equivalente à exibida na figura 6.5.

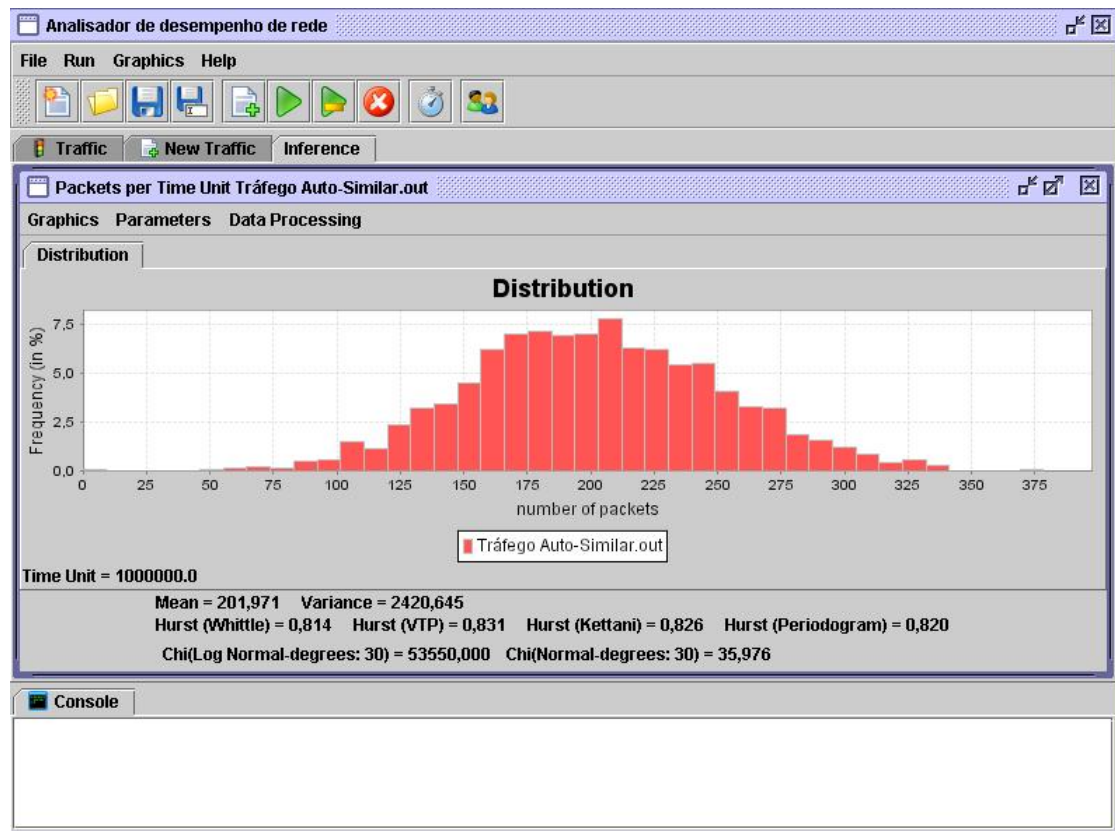


Figura 6.5 – Janela: ‘Inferência’ – Gráfico da distribuição do processo

Nessa janela é possível a visualização de vários tipos de gráficos referentes às estatísticas do processo, como por exemplo, a distribuição e o *variance time plot*, através do menu ‘*Graphics*’. Ainda é possível o cálculo de parâmetros do processo, como por exemplo a média, a variância e o parâmetro de Hurst, através do menu ‘*Parameters*’. O último menu, ‘*Data Processing*’, contém a opção ‘*Remove Trend of Degree 10*’ que torna possível a remoção do *trend* como mostrado no capítulo 7.

7. Resultados Obtidos

Nesse capítulo, apresentamos os resultados que obtivemos utilizando as medidas de inferência para analisarmos a eficácia de alguns modelos de geração de tráfego descritos anteriormente e para tentarmos determinar um modelo para o tráfego de uma rede real. Além disso, propomos um modelo simples para o nó de uma rede e verificamos sua resposta a diferentes tipos de tráfego no que diz respeito aos principais parâmetros de QoS: atraso e perda de pacotes.

7.1. Testes de Geração

Fizemos dois tipos de geração de tráfego: o primeiro para testarmos a eficácia da geração de um tráfego baseado em variáveis independentes com uma distribuição conhecida e o segundo para analisarmos se o tráfego auto-similar gerado possui as características de auto-similaridade esperada.

7.1.1. Geração de Tráfego Poisson

Nesse teste utilizamos o método da inversão para gerarmos variáveis aleatórias independentes com uma certa p.d.f. e geramos o tráfego baseado nessas variáveis. Escolhemos gerar uma distribuição exponencial com média 0,1 para representar os tempos entre os pacotes em segundos. Isso foi feito porque o tráfego gerado dessa forma é um tráfego Poisson, que foi o primeiro modelo de tráfego utilizado para redes de dados.

A figura abaixo mostra o histograma dos tempos entre pacotes verificados nessa geração.

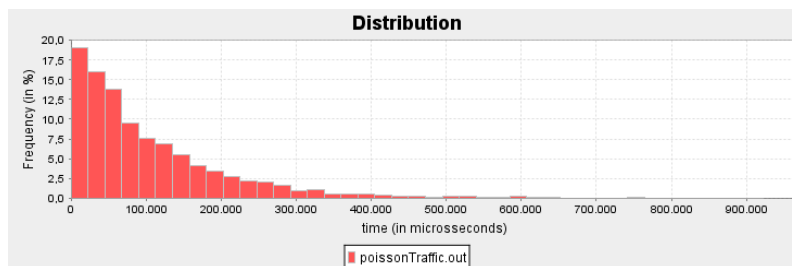


Figura 7.1 – Histograma do tráfego de Poisson gerado

Pela figura 1 percebemos que o tráfego gerado passa no teste visual. Além disso, calculamos o valor do qui-quadrado para esses tempos. O único valor de qui-quadrado aceitável foi o qui-quadrado para a distribuição exponencial, que foi $\chi^2 = 31,784$, o que indica que a distribuição deve ser mesmo do tipo exponencial. A média calculada para esses tempos foi de 0,103 segundos. Isso mostra que o tráfego gerado possui as características estatísticas desejadas.

7.1.2. Geração de Tráfego Auto-similar

Para realizar esse teste, utilizamos o método definido na seção 5.2 para gerarmos tráfegos auto-similares com taxa média de 20 pacotes por segundo e com parâmetro de Hurst variando entre 0,65 e 0,9. Então utilizamos os estimadores definidos na seção 4.3 para verificarmos os parâmetros de Hurst de cada um dos tráfegos gerados. Os valores estimados para cada caso estão mostrados na tabela 7.1.

Tabela 7.1 – Estimação de parâmetros dos tráfegos fGn gerados

Valores Esperados		Valores Observados				
Média	Hurst	Média	Hurst (Whittle)	Hurst (VTP)	Hurst (Periodograma)	Hurst (Kettani-Gubner)
20	0.65	19.995	0.654	0.653	0.658	0.654
20	0.70	20.120	0.699	0.700	0.712	0.698
20	0.75	19.953	0.750	0.751	0.763	0.749
20	0.80	20.202	0.791	0.787	0.808	0.788
20	0.85	20.351	0.848	0.830	0.848	0.837
20	0.90	20.743	0.894	0.880	0.912	0.882

A tabela 7.1 mostra que o tráfego gerado apresenta realmente as características de auto-similaridade esperadas.

7.2. Estimação de Tráfegos Reais

Como explicado na seção 4.3, para estimarmos o comportamento de um processo real, devemos realizar um pré-processamento para remover possíveis *trends* e periodicidade, que podem afetar a estimação. Uma forma muito utilizada para a remoção dessas componentes indesejadas é passarmos o sinal por um filtro passa-altas.

No entanto, estamos interessados no parâmetro de Hurst, principalmente em sua influência em grandes escalas, ou seja, na dependência de longa duração do processo. Por isso, devemos avaliar o comportamento do sinal em grandes escalas, o que significa avaliá-lo em baixas frequências, impossibilitando o uso de um filtro passa-baixas para remoção dos *trends* e da periodicidade. O artigo [13] propõe três outros métodos muito utilizados para realizar esse pré-processamento, que são:

- Realizar o logaritmo nos dados originais (apropriado somente quando os dados são positivos)
- Remoção do *trend* linear (retirar a reta que melhor se ajusta aos dados)
- Remoção do polinômio de décima ordem que melhor se ajusta aos dados

O método que utilizamos é o método da remoção do polinômio de décima ordem. Apesar desses métodos serem capazes de remover o *trend* do sinal, muitas vezes eles não são capazes de remover a periodicidade.

Os dados utilizados nessa seção correspondem ao tráfego médio do *backbone* de uma grande operadora de Internet calculado em uma janela de 5 minutos. Esses dados foram modificados neste trabalho por um fator, o que não modifica as características estatísticas que desejamos inferir. A figura 7.2 ilustra o tráfego obtido (modificado por um fator).

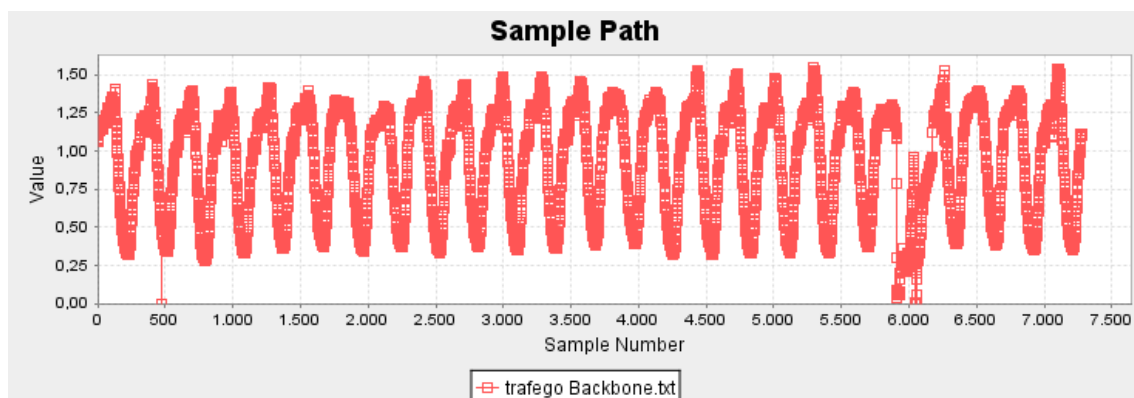


Figura 7.2 – Tráfego de um backbone

Nós também conseguimos obter os dados de tráfego separados por tipos de tráfego. As figuras 7.3 e 7.4 ilustram dois tipos de tráfego diferentes, denominados neste trabalho de app1 e app2.

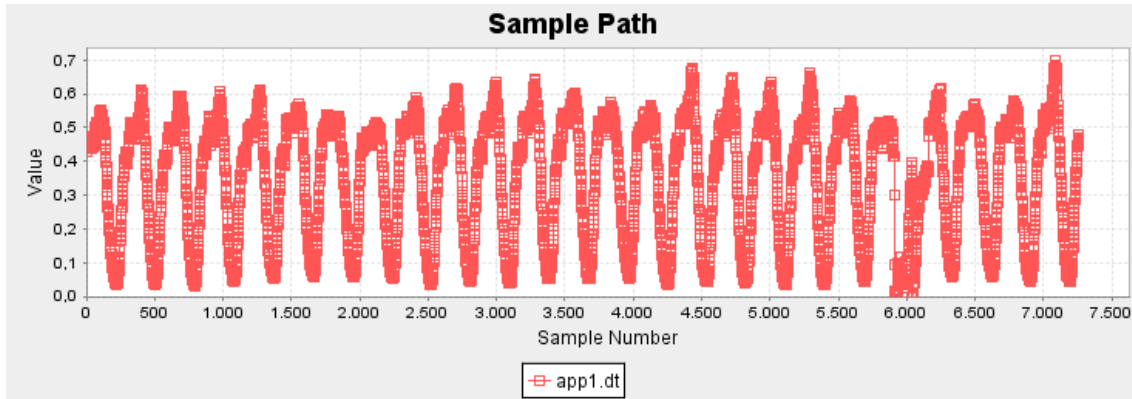


Figura 7.3 – Tráfego de um backbone referente a app1

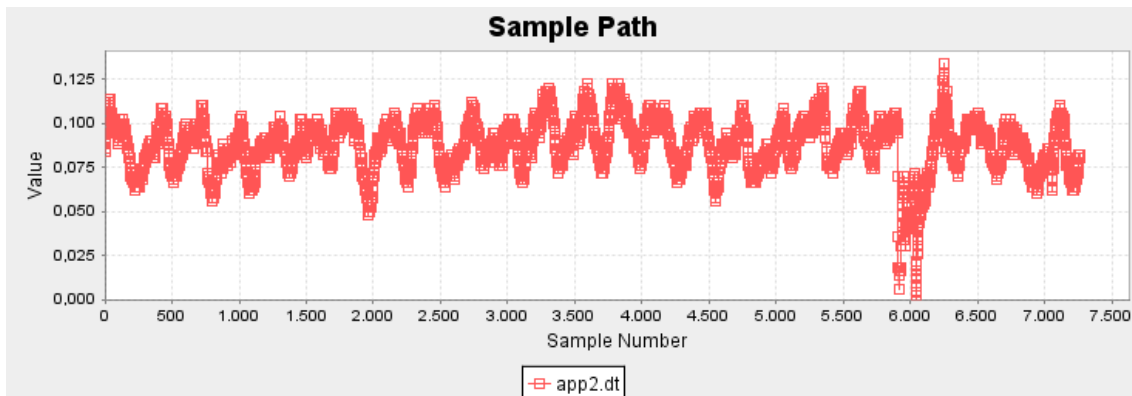


Figura 7.4 – Tráfego de um backbone referente a app2

Podemos perceber claramente pelas figuras que os tráfegos medidos possuem uma componente periódica além do *trend*. A componente periódica advém do padrão diário de uso da rede. O *trend* existe porque a taxa média de utilização da rede muda com o tempo. Como os métodos estabelecidos anteriormente não removem a componente periódica, nós analisamos os tráfegos somente pelo período de um dia, pois nesse período, o tráfego não possuirá periodicidade. Para ilustrar a ineficiência dos métodos anteriores em retirar mostramos na figura 7.5 o tráfego do *backbone* após retirarmos o polinômio de décima ordem que mais se ajusta a esse tráfego. Na figura 7.6 mostramos o tráfego do *backbone* no período de um dia e na figura 7.7 esse tráfego após removermos o *trend* utilizando o método do polinômio de décima ordem.

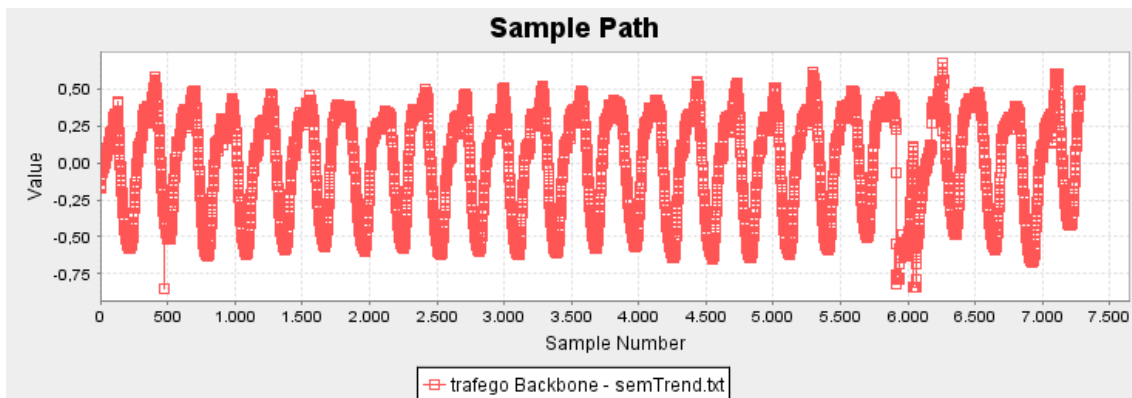


Figura 7.5 – Tráfego do backbone após aplicação do método de remoção de *trend*

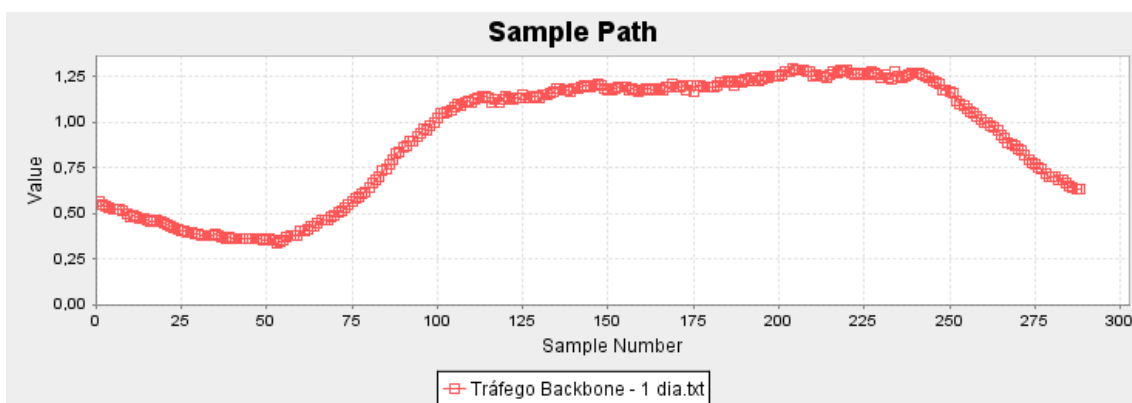


Figura 7.6 – Tráfego do backbone por 1 dia

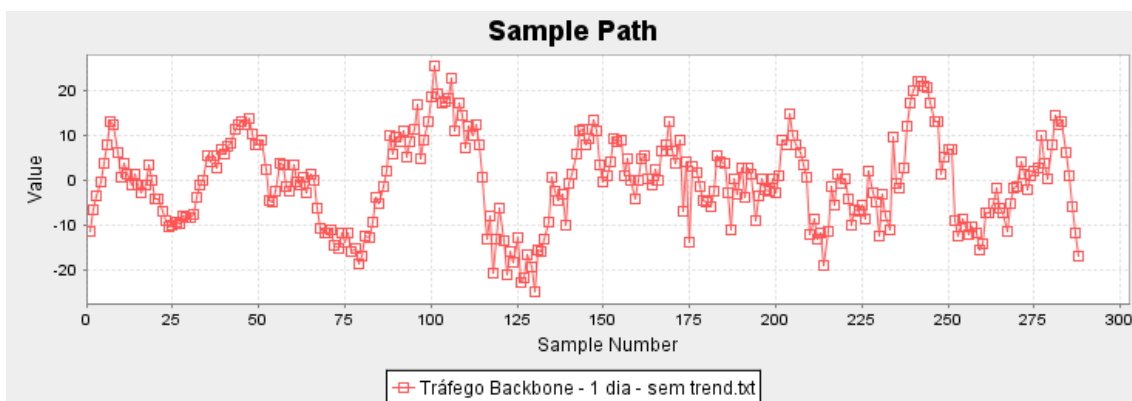


Figura 7.7 – Tráfego do backbone por 1 dia depois da remoção do *trend*

Com os dados obtidos, nós analisamos a auto-similaridade e a distribuição do tráfego para 17 dias diferentes. Na tabela 7.2, mostramos os valores do parâmetro de Hurst para cada dia de tráfego, obtidos por dois estimadores diferentes, e os valores de qui-quadrado calculados para uma distribuição Gaussiana. Os valores de qui-quadrado

obtidos para as outras distribuições foram inaceitáveis. Além disso, as tabelas 7.3 e 7.4 mostram as mesmas estimativas para app1 e app2. Os valores dessa tabela mostram que, na escala avaliada (5 minutos), o tráfego no *backbone*, realmente possui, características auto-similares e seguem uma distribuição gaussiana, o que indica que o tráfego deve ser, pelo menos assintoticamente, um processo fGn.

Tabela 7.2 – Estimação de parâmetros estatísticos do tráfego de backbone

Dia	Hurst (<i>Variance-Time Plot</i>)	Hurst (Kettani-Gubner)	Qui-Quadrado (Distribuição Gaussiana)
1	0,928	0,965	47,618
2	0,951	0,976	70,653
3	0,944	0,967	51,924
4	0,916	0,943	37,069
5	0,91	0,943	16,403
6	0,911	0,941	26,306
7	0,955	0,974	42,451
8	0,945	0,973	31,903
9	0,955	0,979	46,972
10	0,939	0,964	48,694
11	0,887	0,944	21,354
12	0,916	0,947	52,785
13	0,954	0,978	35,132
14	0,947	0,97	51,924
15	0,948	0,973	45,25
16	0,951	0,975	23,722
17	0,938	0,966	38,146

Tabela 7.3 – Estimação de parâmetros estatísticos de app1

Dia	Hurst (<i>Variance-Time Plot</i>)	Hurst (Kettani-Gubner)	Qui-Quadrado (Distribuição Gaussiana)
1	0,843	0,895	31,042
2	0,812	0,878	71,299
3	0,901	0,926	38,146
4	0,9	0,934	52,569
5	0,815	0,879	32,549
6	0,816	0,904	62,042
7	0,865	0,906	17,91
8	0,87	0,916	39,653
9	0,907	0,935	28,028
10	0,867	0,919	21,785
11	0,869	0,906	27,167
12	0,671	0,861	35,778
13	0,878	0,909	36,208
14	0,839	0,894	44,604
15	0,874	0,907	30,611
16	0,753	0,85	23,292
17	0,851	0,914	40,299

Tabela 7.4 – Estimação de parâmetros estatísticos de app2

Dia	Hurst (<i>Variance-Time Plot</i>)	Hurst (Kettani-Gubner)	Qui-Quadrado (Distribuição Gaussiana)
1	0,915	0,948	63,549
2	0,942	0,962	49,771
3	0,937	0,963	45,25
4	0,902	0,935	28,243
5	0,902	0,928	20,708
6	0,901	0,942	30,181
7	0,939	0,964	43,258
8	0,932	0,964	52,354
9	0,937	0,968	39,007
10	0,922	0,948	38,576
11	0,86	0,926	37,5
12	0,904	0,942	33,84
13	0,937	0,965	55,799
14	0,922	0,958	46,972
15	0,935	0,963	46,757
16	0,935	0,967	49,986
17	0,933	0,964	37,285

7.3. Simulação de Fila

Para simularmos a resposta de um nó a um determinado tipo de tráfego, utilizamos um modelo simples para o atraso inserido em um nó da rede, consistindo basicamente da soma do atraso médio de processamento, da espera do pacote no buffer e do tempo de transmissão.

Baseando-nos nesse modelo de nó, realizamos quatro simulações e verificamos a distribuição do atraso e a taxa de perda de pacotes para cada uma das simulações. As simulações foram feitas com os tamanhos do pacote distribuídos exponencialmente com média de 250 bytes, tamanho do buffer de 256 MB, capacidade do enlace de 10Mbps, e com taxa média de chegada de 7Mbps.

A primeira simulação foi feita com o processo de chegada sendo um processo de Poisson. Na segunda, terceira e quarta simulações, fizemos o processo de chegadas sendo um fGn com parâmetros de Hurst iguais a 0,5, 0,7 e 0,9. Dessa forma, obtemos as distribuições do atraso para cada um dos casos, ilustradas pelas figuras abaixo.

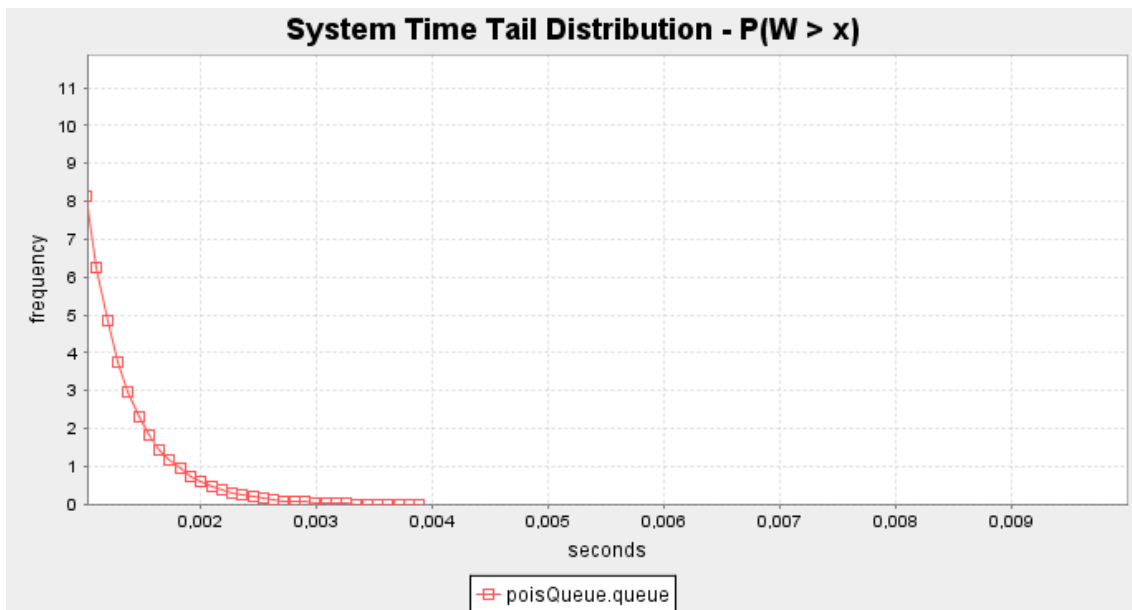


Figura 7.8 – Distribuição de cauda do atraso em um nó para o tráfego Poisson

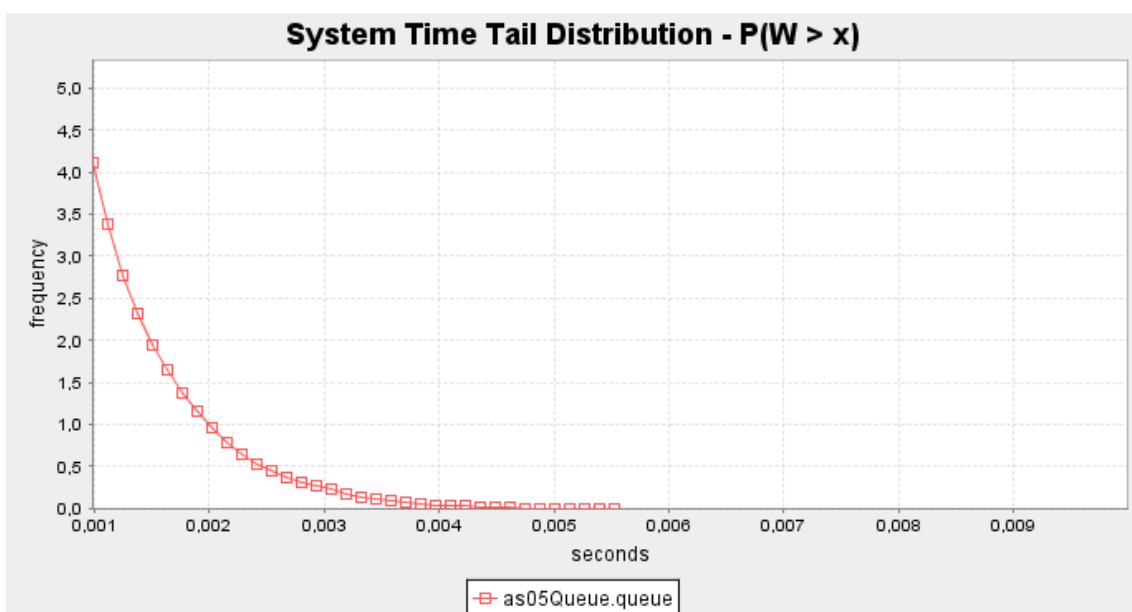


Figura 7.9 – Distribuição de cauda do atraso do tráfego auto-similar com parâmetro de Hurst igual a 0,5

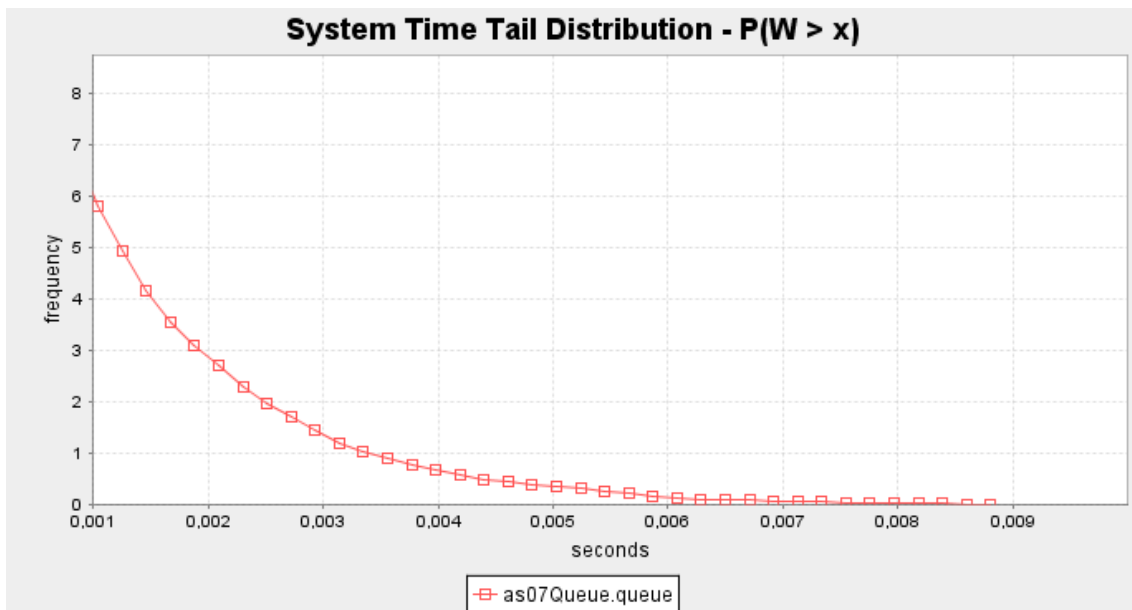


Figura 7.9 – Distribuição de cauda do atraso do tráfego auto-similar com parâmetro de Hurst igual a 0,7

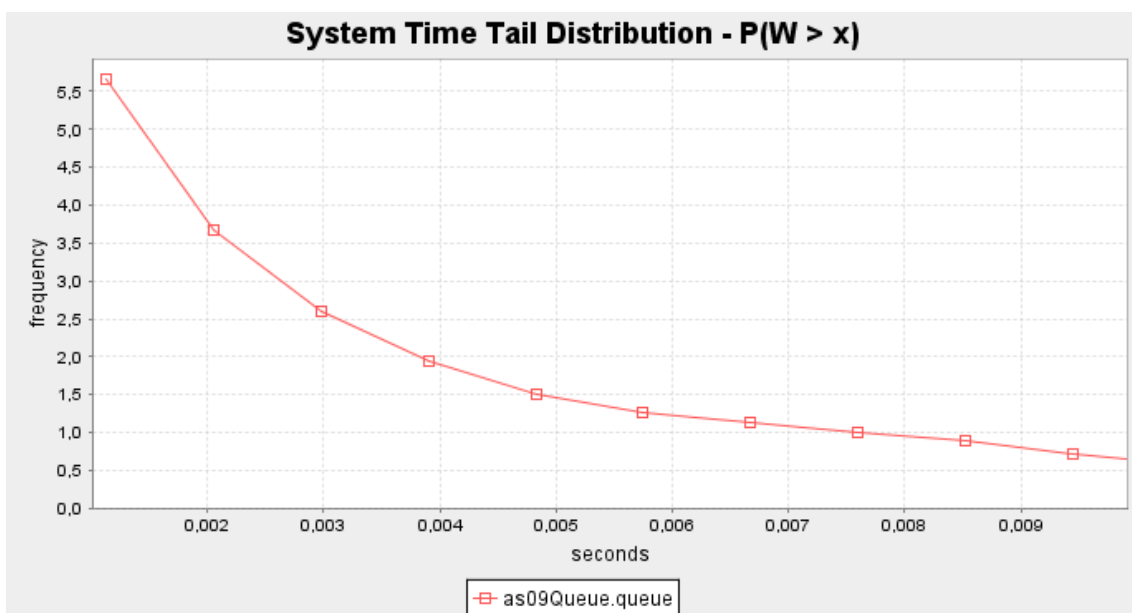


Figura 7.10 – Distribuição de cauda do atraso do tráfego auto-similar com parâmetro de Hurst igual a 0,9

Nessas condições a taxa de perda de pacotes em todos os casos foi de 0%. As figuras mostram que a probabilidade do atraso ser grande em um tráfego auto-similar é maior que em um tráfego Poisson. Além disso, vemos que quanto maior for o parâmetro de Hurst, maior será essa probabilidade. Logo, para garantirmos um mesmo atraso

máximo para um tráfego de Poisson são exigidos menos recursos que para um tráfego auto-similar.

8. Conclusão

Percebemos que tráfegos reais do *backbone* de uma rede possuem características auto-similares, pelo menos para escalas grandes (neste trabalho avaliamos os tráfegos reais apenas em escalas maiores que 5 minutos). As duas classes de tráfego analisadas também possuem características auto-similares. Além disso, os resultados obtidos na seção 7.1 mostram que os métodos implementados geram os processos estocásticos propostos com boa aderência. Por fim, a simulação de fila realizada mostra que a característica auto-similar do tráfego possui uma influência direta na qualidade dos serviços oferecidos pela rede. E, devido à dificuldade na inferência das características estatísticas dos tráfegos reais, é de grande importância que projetistas de rede tenham uma ferramenta que os auxilie no correto dimensionamento de redes multiserviços.

9. Trabalhos Futuros

Nesse trabalho, analisamos as características estatísticas dos tráfegos reais para grandes escalas, devido à dificuldade da obtenção de dados de tráfego para pequenas escalas. Como trabalhos futuros, devemos avaliar o comportamento do tráfego também em escalas pequenas e utilizarmos modelos que contenham informações do tráfego em várias escalas (como o MWM) para avaliarmos a resposta da rede a diferentes tipos de tráfego, de forma a considerar suas características tanto para escalas pequenas quanto para escalas grandes.

APÊNDICES

A Tráfego Poisson

Devido ao sucesso da utilização do processo de Poisson em redes de telefonia para fazer a modelagem do tempo entre a chegada de novas chamadas e a duração dessas, ele foi o primeiro utilizado para caracterizar o tráfego que chega em um nó de uma rede de computadores. Nesse capítulo derivaremos o processo de Poisson e verificaremos algumas de suas propriedades.

A.1 Processo de Poisson

O processo de Poisson é um processo de chegadas $\{N(t), t > 0\}$, em que $N(t)$ representa o número de chegadas ocorridas até o tempo t (com $N(0) = 0$), e que satisfaz as seguintes condições:

- i. A probabilidade de ocorrer alguma chegada entre o tempo t e $t + \Delta t$ é:
 $P\{\text{chegada entre } t \text{ e } \Delta t\} = \lambda \Delta t + o(\Delta t)$, em que λ é uma constante independente de $N(t)$ e $o(\Delta t)$ representa uma quantidade pequena comparada a Δt quando Δt tende a zero. Ou seja:

$$\lim_{\Delta t \rightarrow 0} \frac{o(\Delta t)}{\Delta t} = 0.$$

- ii. $P\{\text{mais de uma chegada entre } t \text{ e } \Delta t\} = o(\Delta t)$.
- iii. O número de chegadas em intervalos disjuntos é independente, ou seja, o processo possui incrementos independentes.

A partir dessas condições, vamos calcular $p_n(t)$, a probabilidade da ocorrência de n chegadas entre 0 e t , ou seja, vamos calcular a p.d.f. do processo $N(t)$. Para fazer isso vamos primeiro desenvolver as equações diferenciais do processo. Para $n \geq 1$ temos:

$$\begin{aligned} p_n(t + \Delta t) = & P\{n \text{ chegadas em } t \text{ e nenhuma em } \Delta t\} + \\ & P\{n - 1 \text{ chegadas em } t \text{ e } 1 \text{ em } \Delta t\} + \\ & P\{n - 2 \text{ chegadas em } t \text{ e } 2 \text{ em } \Delta t\} + \dots + \\ & P\{0 \text{ chegadas em } t \text{ e } n \text{ em } \Delta t\}. \end{aligned} \tag{A.1}$$

A partir das condições podemos calcular as probabilidades $P\{n-j \text{ chegadas em } t \text{ e } j \text{ em } \Delta t\}$:

Para $j = 0$

$$P\{n \text{ chegadas em } t \text{ e nenhuma em } \Delta t\} = P\{n \text{ chegadas em } t\}P\{0 \text{ chegadas em } \Delta t\} = p_n(t)P\{0 \text{ chegadas em } \Delta t\}$$

$$P\{0 \text{ chegadas em } \Delta t\} = 1 - P\{\text{chegada em } \Delta t\} = 1 - \lambda\Delta t - o(\Delta t)$$

$$P\{n \text{ chegadas em } t \text{ e nenhuma em } \Delta t\} = p_n(t)[1 - \lambda\Delta t - o(\Delta t)]$$

Para $j = 1$

$$P\{n-1 \text{ chegadas em } t \text{ e } 1 \text{ em } \Delta t\} = P\{n-1 \text{ chegadas em } t\}P\{1 \text{ chegada em } \Delta t\} = p_{n-1}(t)P\{1 \text{ chegada em } \Delta t\}$$

$$P\{\text{uma chegada em } \Delta t\} = \lambda\Delta t + o(\Delta t)$$

$$P\{n-1 \text{ chegadas em } t \text{ e } 1 \text{ em } \Delta t\} = p_{n-1}(t)[\lambda\Delta t + o(\Delta t)]$$

Para $j \geq 2$:

$$P\{n-j \text{ chegadas em } t \text{ e } j \text{ em } \Delta t\} = P\{n-j \text{ chegadas em } t\}P\{j \text{ chegadas em } \Delta t\} = p_{n-j}(t)P\{j \text{ chegadas em } \Delta t\}$$

$$P\{j \text{ chegadas em } \Delta t\} = P\{\text{mais de uma chegada em } \Delta t\} = o(\Delta t)$$

$$P\{n-j \text{ chegadas em } t \text{ e } j \text{ em } \Delta t\} = p_{n-j}(t)o(\Delta t)$$

$$\lim_{\Delta t \rightarrow 0} \frac{p_{n-j}(t)o(\Delta t)}{\Delta t} = p_{n-j}(t) \lim_{\Delta t \rightarrow 0} \frac{o(\Delta t)}{\Delta t} = 0 \Rightarrow p_{n-j}(t)o(\Delta t) \text{ é uma função do tipo } o(\Delta t)$$

$$P\{n-j \text{ chegadas em } t \text{ e } j \text{ em } \Delta t\} = o(\Delta t)$$

Além disso, a soma de todas as probabilidades $P\{n-j \text{ chegadas em } t \text{ e } j \text{ em } \Delta t\}$ para $j \geq 2$ é uma função do tipo $o(\Delta t)$, já que a soma de várias funções do tipo $o(\Delta t)$ resulta em uma função do tipo $o(\Delta t)$.

Se $f_1(\Delta t), f_2(\Delta t), \dots, f_n(\Delta t)$. São funções do tipo $o(\Delta t)$ então:

$$\lim_{\Delta t \rightarrow 0} \frac{f_1(\Delta t)}{\Delta t} = 0, \lim_{\Delta t \rightarrow 0} \frac{f_2(\Delta t)}{\Delta t} = 0, \dots, \lim_{\Delta t \rightarrow 0} \frac{f_n(\Delta t)}{\Delta t} = 0.$$

$$\lim_{\Delta t \rightarrow 0} \frac{f_1(\Delta t) + f_2(\Delta t) + \dots + f_n(\Delta t)}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{f_1(\Delta t)}{\Delta t} + \lim_{\Delta t \rightarrow 0} \frac{f_2(\Delta t)}{\Delta t} + \dots + \lim_{\Delta t \rightarrow 0} \frac{f_n(\Delta t)}{\Delta t} = 0$$

$$g(\Delta t) = f_1(\Delta t) + f_2(\Delta t) + \dots + f_n(\Delta t) \text{ é uma função } o(\Delta t)$$

Então, para $n \geq 1$, temos que:

$$p_n(t + \Delta t) = p_n(t)[1 - \lambda \Delta t - o(\Delta t)] + p_{n-1}(t)[\lambda \Delta t + o(\Delta t)] + o(\Delta t) \quad (n \geq 1) \quad (\text{A.2})$$

Para o caso de $n = 0$ temos que:

$$p_0(t + \Delta t) = p_0(t)[1 - \lambda \Delta t - o(\Delta t)] \quad (\text{A.3})$$

Podemos reescrever as equações (A.2) e (A.3) da seguinte forma:

$$p_0(t + \Delta t) - p_0(t) = -p_0(t)\lambda \Delta t - o(\Delta t)p_0(t) = -p_0(t)\lambda \Delta t + o(\Delta t) \quad (\text{A.4})$$

$$p_n(t + \Delta t) - p_n(t) = -p_n(t)\lambda \Delta t + p_{n-1}(t)\lambda \Delta t + o(\Delta t) \quad (n \geq 1) \quad (\text{A.5})$$

Dividindo as equações (A.4) e (A.5) por Δt e tirando o limite para $\Delta t \rightarrow 0$, obtemos:

$$\lim_{\Delta t \rightarrow 0} \left[\frac{p_0(t + \Delta t) - p_0(t)}{\Delta t} = -p_0(t)\lambda + \frac{o(\Delta t)}{\Delta t} \right]$$

$$\lim_{\Delta t \rightarrow 0} \left[\frac{p_n(t + \Delta t) - p_n(t)}{\Delta t} = -p_n(t)\lambda + p_{n-1}(t)\lambda + \frac{o(\Delta t)}{\Delta t} \right] \quad (n \geq 1)$$

$$\frac{dp_0(t)}{dt} = -\lambda p_0(t) \quad (\text{A.6})$$

$$\frac{dp_n(t)}{dt} = -\lambda p_n(t) + \lambda p_{n-1}(t) \quad (n \geq 1) \quad (\text{A.7})$$

A equação (A.6) tem a solução geral dada por $p_0(t) = Ce^{-\lambda t}$. Como $p_0(0) = 1$, temos que $C = 1$. Logo, $p_0(t) = e^{-\lambda t}$.

Para $n = 1$ temos que:

$$\begin{aligned}\frac{dp_1(t)}{dt} &= -\lambda p_1(t) + \lambda p_0(t) \\ \frac{dp_1(t)}{dt} + \lambda p_1(t) &= \lambda p_0(t) = \lambda e^{-\lambda t}\end{aligned}\tag{A.8}$$

A solução de (A.8) é $p_1(t) = Ce^{-\lambda t} + \lambda t e^{-\lambda t}$. Como $p_1(0) = 0$ para qualquer $n \geq 1$, temos que $C = 0$ e $p_1(t) = \lambda t e^{-\lambda t}$.

Para $n = 2$:

$$\begin{aligned}\frac{dp_2(t)}{dt} &= -\lambda p_2(t) + \lambda p_1(t) \\ \frac{dp_2(t)}{dt} + \lambda p_2(t) &= \lambda p_1(t) = \lambda^2 t e^{-\lambda t}\end{aligned}\tag{A.9}$$

A solução geral de (A.9) é $p_2(t) = Ce^{-\lambda t} + \frac{\lambda^2 t^2}{2} e^{-\lambda t}$, e a solução específica é

$p_2(t) = \frac{\lambda^2 t^2}{2} e^{-\lambda t}$. Resolvendo para $n = 3$ obtemos $p_3(t) = \frac{(\lambda t)^3}{3!} e^{-\lambda t}$. Percebemos que

para todos os casos anteriores

$$p_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}\tag{A.10}$$

De fato, podemos provar (A.10) por indução provando duas coisas:

$$p_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t} \Rightarrow p_{n+1}(t) = \frac{(\lambda t)^{n+1}}{(n+1)!} e^{-\lambda t} \quad (i)$$

e

$$p_0(t) = \frac{(\lambda t)^0}{0!} e^{-\lambda t} \quad (ii)$$

Nós já provamos (ii) quando calculamos $p_0(t)$. Provamos (i) da seguinte forma:

$$\frac{dp_{n+1}(t)}{dt} + \lambda p_{n+1}(t) = \lambda p_n(t) = \frac{\lambda^{n+1} t^n}{n!} e^{-\lambda t}$$

$$\frac{dp_{n+1}(t)}{dt} e^{\lambda t} + \lambda e^{\lambda t} p_{n+1}(t) = \frac{\lambda^{n+1} t^n}{n!} e^{-\lambda t} e^{\lambda t} = \frac{\lambda^{n+1} t^n}{n!}$$

$$\frac{d}{dt} (p_{n+1}(t) e^{\lambda t}) = \frac{\lambda (\lambda t)^n}{n!} \Rightarrow p_{n+1}(t) e^{\lambda t} = \int \frac{\lambda^{n+1} s^n}{n!} ds + C \Rightarrow p_{n+1}(t) = e^{-\lambda t} \frac{\lambda^{n+1}}{n!} \int s^n ds + C e^{-\lambda t}$$

$$p_{n+1}(t) = e^{-\lambda t} \frac{\lambda^{n+1}}{n!} \frac{t^{n+1}}{n+1} + C e^{-\lambda t} = \frac{(\lambda t)^{n+1}}{(n+1)!} + C e^{-\lambda t}$$

$$p_n(0) = 0 \text{ para } n > 0 \Rightarrow C = 0$$

$$p_{n+1}(t) = \frac{(\lambda t)^{n+1}}{(n+1)!}$$

A fórmula dada por (A.10) representa uma p.d.f. poissoniana com média λt . Logo, um processo de Poisson, definido pelas condições i, ii e iii, possui p.d.f. com uma média de λt chegadas, ou seja, uma taxa média de λ chegadas por unidade de tempo.

A.2 Propriedades do Processo de Poisson

Um processo de Poisson possui várias propriedades interessantes, dentre as quais:

- a. Possui incrementos estacionários, ou seja, as ocorrências em intervalos de mesmo tamanho são identicamente distribuídas.
- b. Em processo de chegadas de Poisson, a variável aleatória associada ao tempo entre as chegadas segue uma distribuição exponencial.

Para provar a afirmação (a) devemos mostrar que, para um processo de Poisson, $N(t) - N(s)$ possui a mesma distribuição que $N(t + h) - N(s + h)$. Isso pode ser facilmente verificado se notarmos que, já que os incrementos são independentes, podemos considerar, sem perda de generalidade, $N(s)$ e $N(s + h)$ como sendo zero. Assim, se derivarmos a p.d.f. de Poisson para $N(t)$ e $N(t + h)$ da mesma forma que anteriormente, obtemos a seguinte distribuição para ambos os casos:

$$p_n(t - s) = \frac{(\lambda(t - s))^n}{n!} e^{-\lambda(t - s)}$$

Vamos agora demonstrar que o tempo entre chegadas deve seguir uma distribuição exponencial. Se T for a variável aleatória que representa o tempo entre pacotes temos que:

$$P\{T \geq t\} = P\{\text{não chegar nada até } t\} = p_0(t) = e^{-\lambda t}$$

Logo a CDF de T é dada por: $C(t) = P\{T \leq t\} = 1 - e^{-\lambda t}$, que é a CDF de uma variável com média $1/\lambda$. Isso vai de acordo com o esperado. Se a taxa média é λ , o tempo médio entre os pacotes deve ser $1/\lambda$.

A.3 Propriedade “Markoviana” da Distribuição Exponencial

A distribuição exponencial é a única distribuição continua que possui a propriedade de ser sem memória. Essa propriedade se refere ao fato de que a probabilidade do tempo restante ser de t unidades de tempo independe de quanto tempo já se passou, ou seja:

$$P\{T \leq t_1 \mid T \geq t_0\} = P\{0 \leq T \leq t_1 - t_0\}$$

A prova de que a distribuição exponencial possui essa propriedade é simples. A partir da definição de probabilidade condicional temos que:

$$P\{T \leq t_1 \mid T \geq t_0\} = \frac{P\{T \leq t_1, T \geq t_0\}}{P\{T \geq t_0\}} = \frac{e^{-\lambda t_0} - e^{-\lambda t_1}}{e^{-\lambda t_0}} = 1 - e^{-\lambda(t_1 - t_0)} = P\{0 \leq T \leq t_1 - t_0\}$$

B Tráfego Rajada

Também conhecido como tráfego ON/OFF ou “Burst”, é o modelo mais simples para modelar um tráfego de voz. O estado OFF é o período de silêncio, ou seja, o período do sistema que não é utilizado pois não há atividade de voz. E o estado ON representa a fonte emissora do tráfego no período ativo da voz. O tempo médio em cada estado é dado por uma distribuição exponencial, no estado ON com média $1/\mu$ e no estado OFF com média $1/\lambda$. Dessa forma, a transição OFF→ON ocorre a uma taxa λ e a transição ON→OFF ocorre a uma taxa μ , conforme a figura B.1.

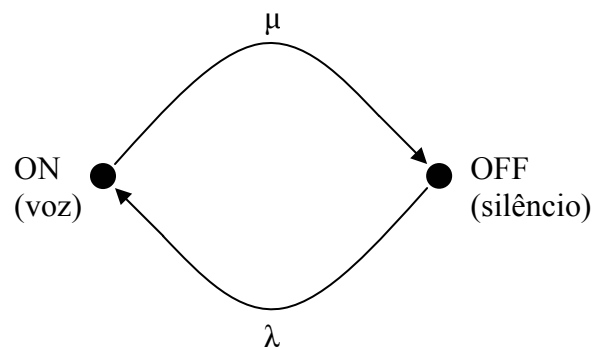


Figura B.1 – Modelo Rajada para uma única fonte de voz.

Durante o período ON a taxa de chegadas de informações é constante, portanto o tempo entre chegadas é determinístico.

Esse modelo foi utilizado e com muito sucesso na modelagem de voz principalmente por duas razões:

- i) experimentos atuais com uma única fonte de voz têm mostrado que durante uma conversação a distribuição dos períodos de silêncio e de atividade da voz podem ser representados por distribuições exponenciais e
- ii) o modelo de fonte única pode ser expandido para um processo de nascimento e morte multi-estado para representar voz multiplexada.

Para tráfego de dados o modelo rajada também pode ser utilizado com sucesso para modelar transmissões de dados que ocorrem com períodos de atividade seguidos por períodos de inatividade. O modelo IPP (*Interrupted Poisson Process*) [14] é um caso especial desse processo, onde os períodos ON/OFF e o tempo entre chegadas de pacotes durante o período ON são mutuamente independentes e todos seguem uma distribuição exponencial. De acordo com [3], fontes ON-OFF podem ser utilizadas sozinhas ou em conjunto para modelar diversos tráfegos, como mostra a Tabela B.1.

Tabela B.1 – Possibilidades de modelagem utilizando fontes rajada.

Número de fontes em paralelo	Distribuição do tempo entre chegadas	Distribuição dos períodos ON e OFF	Tipo de Aplicação	Denominação do modelo	Exemplo de aplicação
4	Exponencial	Exponencial	Dados	4IPP	WAN
1	Constante	Pareto	Dados	Pareto ON-OFF	LAN
1	Weibull	Pareto	Dados	WWW	PAN
1	Constante	Exponencial	Voz	ON-OFF	Telefone
1	Exponencial	Pareto	Vídeo	Pareto IPP	DTV
16	Constante	Exponencial	Vídeo	Mini ON-OFF	Vídeo Conferência

Em [2] é provado que é possível gerar um tráfego auto-similar a partir de um grande número de fontes ON-OFF em paralelo desde que o período ON e o OFF possuam duração com variância muito grande ou infinita. Não foi possível incluir esse método de geração de tráfego na nossa ferramenta, pois necessitaria de um processador de grande capacidade ou de um grande conjunto de processadores em paralelo para simular as diversas fontes. No experimento realizado em [2] é utilizado um computador com 16.384 processadores em paralelo para gerar tráfegos com um grande número de fontes.

Fontes ON-OFF podem também ser utilizadas para simular tráfegos de dados em redes sem fio com múltiplo acesso por divisão de tempo, onde uma transmissão é realizada em uma janela de transmissão, permanecendo inativo nas janelas destinadas a outros usuários.

C Teoria de Filas

Por definição, fila é uma série de indivíduos ou de tarefas, que são dispostos em uma ordem predefinida, para que se possa realizar um determinado. A teoria de filas é utilizada para prever o comportamento de uma fila que exista, ou para projetar uma fila que se deseja implementar. No caso de um nó de uma rede, a análise é feita sobre a fila de pacotes que se forma devido à taxa de chegada de pacotes em um servidor ou roteador, ser maior que a capacidade de processamento dessa estação ou da capacidade do link de saída em um determinado espaço de tempo.

C.1 Notação de Kendall

Em uma fila há algumas características que são utilizadas para denominá-la: taxa de chegada de usuários à fila, tempo médio gasto no atendimento dos usuários enfileirados, quantidade de atendentes que processam os pedidos dos usuários da fila, tamanho máximo de usuários que a fila suporta e disciplina de atendimento da fila. O termo atendente ou servidor em um sistema de filas é equivalente. Na teoria de filas é usada uma notação para caracterizar uma fila de acordo com esses parâmetros, que é conhecida como notação de Kendall, essa notação utiliza uma série de símbolos e barras da forma $A/B/X/Y/Z$, onde A indica a distribuição do tempo entre chegadas, B a distribuição de probabilidade para o tempo de serviço, X o número de servidores em paralelo, Y o limite de capacidade da fila, e Z a disciplina da fila. Alguns dos símbolos padrões utilizado para essas características estão na Tabela C.1.

Tabela C.1 – Notação A/B/X/Y/Z

Característica	Símbolo	Explicação
Distribuição do tempo entre chegadas (A) Distribuição do tempo de serviço (B)	M	Exponencial
	D	Determinístico
	E_k	Erlang tipo k ($k = 1, 2, \dots$)
	H_k	Mistura de k exponenciais
	PH	Tipo Fase
	G	Geral – não especificado
Número de servidores em paralelo (X)	$1, 2, \dots, \infty$	
Limite de capacidade da fila (Y)	$1, 2, \dots, \infty$	
Disciplina da fila (Z)	FIFO	Primeiro a chegar, primeiro a ser servido
	LIFO	Último a chegar, primeiro a ser servido
	RSS	Seleção para atendimento randômica
	PR	Prioridade
	G	Geral – não especificado

O símbolo M para exponencial é utilizado pois o uso do símbolo “E” poderia gerar confusão com o símbolo “E_k”, da distribuição Erlang tipo k, e também por indicar a propriedade da distribuição exponencial ser Markoviana e sem memória (*memoryless*).

C.2 Teorema de Little

Uma medida necessária e extremamente importante para a modelagem de uma fila em uma rede de dados é o tempo que um pacote demora em um nó, ou seja, o atraso induzido ao pacote no nó. Para a obtenção dessa medida apresentaremos um resultado muito importante, conhecido como Teorema de Little.

Supondo que observamos um sistema de $t = 0$, até um tempo t indefinido e gravamos alguns valores:

$N(t)$ = Número de clientes no sistema no tempo t

$\alpha(t)$ = Número de clientes que chegaram no intervalo $[0, t]$

T_i = Tempo gasto no sistema pelo i -ésimo cliente

Intuitivamente definimos número médio de clientes observados no sistema até o instante t como:

$$N_t = \frac{1}{t} \int_0^t N(\tau) d\tau$$

Naturalmente, N_t possui valores diferentes para diferentes instantes de tempo t , mas para muitos sistemas de interesse, N_t tende para um valor estacionário N à medida que t aumenta, isto é:

$$N = \lim_{t \rightarrow \infty} N_t \text{ (assumindo que esse limite existe)}$$

Da mesma forma definimos a taxa média de chegadas λ_t , e o valor estacionário λ , entre 0 e t , como:

$$\lambda_t = \frac{\alpha(t)}{t} \quad , \quad \lambda = \lim_{t \rightarrow \infty} \lambda_t \quad (\text{assumindo que esse limite existe})$$

E finalmente o tempo médio de atraso do cliente no sistema T_t , e o valor estacionário T , como:

$$T_t = \frac{\sum_{i=0}^{\alpha(t)} T_i}{\alpha(t)} \quad , \quad T = \lim_{t \rightarrow \infty} T_t \quad (\text{assumindo que esse limite existe})$$

Essas grandezas, N , λ e T são relacionadas por uma fórmula simples que torna possível a obtenção de uma a partir das outras. Essa fórmula é conhecida como Teorema de Little, ou Fórmula de Little em algumas literaturas, que é:

$$N = \lambda T$$

E para o número de usuários na fila e o tempo de espera na fila:

$$N_q = \lambda T_q$$

Outra medida importante em um sistema de fila é a taxa de atendimento μ , que é o inverso do tempo médio gasto por cada atendente desde o momento que recebe o usuário até o momento que o usuário deixa o sistema. Com μ podemos definir uma medida de congestão do sistema, conhecida como ρ , para uma fila G/G/c ρ é definido como:

$$\rho \equiv \frac{\lambda}{c\mu}$$

onde “c” é o número de servidores em paralelo.

Para que o Teorema de Little tenha validade para a análise de uma fila, necessariamente $\rho < 1$, pois se $\rho > 1$ o número médio de chegadas seria maior que o valor máximo de atendimentos e a fila cresceria indefinidamente até o ponto que novos usuários seriam rejeitados e o tamanho da fila nunca diminuiria e nunca estaria em estado estacionário, condição necessária para Little.

A partir de ρ obtemos a probabilidade do sistema estar vazio, sem nenhum usuário:

$$p_0 = 1 - \rho$$

C.3 Modelagem para fila M/M/1

Uma fila M/M/1 é um processo de nascimento e morte markoviano, com $\lambda_n = \lambda$ e $\mu_n = \mu$, para qualquer n . Chegadas podem ser consideradas como “nascimentos” no sistema, se o sistema estiver no estado n e ocorrer uma chegada, o estado será $n+1$. É demonstrado em [10] que:

$$p_{n+1} = \frac{\lambda + \mu}{\mu} p_n - \frac{\lambda}{\mu} p_{n-1} \quad (n \geq 1)$$

$$p_1 = \frac{\lambda}{\mu} p_0 \quad (n \geq 1)$$

E também que:

$$p_n = (1 - \rho) \rho^n \quad (\rho = \lambda/\mu < 1)$$

Com essas probabilidades para o estado estacionário do sistema é possível obter algumas medidas de eficiência, em particular as medidas do número esperado de usuários no sistema e na fila, que podem ser encontrados por:

$$N = \sum_{n=0}^{\infty} n p_n = (1 - \rho) \sum_{n=0}^{\infty} n \rho^n$$

Podem ser realizadas as seguintes operações:

$$\sum_{n=0}^{\infty} n \rho^n = \rho \sum_{n=1}^{\infty} n \rho^{n-1}$$

E como $\sum_{n=1}^{\infty} n\rho^{n-1}$ é a derivada de $\sum_{n=0}^{\infty} \rho^n$ em relação a ρ , e sendo as operações de soma e a derivação intercambiáveis, sendo $\rho < 1$,

$$\sum_{n=0}^{\infty} \rho^n = \frac{1}{1-\rho}$$

Conseqüentemente:

$$\sum_{n=0}^{\infty} n\rho^n = \rho \frac{d[1/(1-\rho)]}{d\rho} = \frac{\rho}{(1-\rho)^2}$$

E finalmente, o número de usuários esperado no sistema em estado estacionário é:

$$N = \frac{\rho(1-\rho)}{(1-\rho)^2} = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}$$

Em [10] é apresentado também o resultado para o número de usuários na fila:

$$N_q = \frac{\rho^2}{1-\rho} = \frac{\lambda^2}{\mu(\mu-\lambda)}$$

Utilizando Little para obter o tempo médio de espera de um usuário no sistema em estado estacionário, temos:

$$T = \frac{N}{\lambda} = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\mu-\lambda}$$

E o tempo médio de espera na fila:

$$T_q = \frac{N_q}{\lambda} = \frac{\rho^2}{\lambda(1-\rho)} = \frac{\rho}{\mu-\lambda}$$

C.4 Modelagem para filas com número de usuários em espera limitado (M/M/1/K)

Para um nó de uma rede processando pacotes, o modelo M/M/1 pode não ser válido pois o buffer onde os pacotes esperam não é infinito e ocorre com frequência desse lotar, causando a perda de pacotes no nó. Assim, é apresentado aqui como obter as medidas de eficiência para esse modelo de filas. Porém é necessário deixar claro que os usuários de um sistema de rede são os pacotes, porém a limitação de usuários é dada em bytes, para o cálculo correto das medidas de eficiência é necessário a conversão de uma das unidades utilizando o tamanho médio do pacote, e ressaltando que a medida será aproximada, pois o tamanho do pacote pode variar com uma certa amplitude.

As medidas de eficiência apresentadas em [10], são:

$$p_0 = \begin{cases} \frac{1-\rho}{1-\rho^{K+1}} \Rightarrow (\rho \neq 1) \\ \frac{1}{K+1} \Rightarrow (\rho = 1) \end{cases}$$

$$p_n = \begin{cases} \frac{(1-\rho)\rho^n}{1-\rho^{K+1}} \Rightarrow (\rho \neq 1) \\ \frac{1}{K+1} \Rightarrow (\rho = 1) \end{cases}$$

$$L_q = \begin{cases} \frac{\rho}{1-\rho} - \frac{\rho(K\rho^K + 1)}{1-\rho^{K+1}} \Rightarrow (\rho \neq 1) \\ \frac{K(K-1)}{2(K+1)} \Rightarrow (\rho = 1) \end{cases}$$

$$N = N_q + (1 - p_0)$$

Nesse modelo a taxa de chegada tem que ser diferenciada pois quando o sistema estiver no estado $n=K$, a taxa de chegadas será igual a 0, portanto:

$$T = \frac{N}{\lambda_{eff}} = \frac{N}{\lambda(1 - p_K)}$$

$$T_q = T - \frac{1}{\mu} = \frac{N_q}{\lambda_{eff}} = \frac{N_q}{\lambda(1-p_K)}$$

C.5 Teorema da Independência de Kleinrock

Em uma rede de dados, a modelagem do processo de chegada de pacotes em um nó da rede se torna uma tarefa muito difícil, pois os pacotes que entram nesse nó são provenientes de nós anteriores que se combinam e se tornam um único tráfego. Além disso, o tempo entre chegadas dos pacotes está extremamente correlacionado com o tamanho dos pacotes, pois como eles provêm de outros nós que possuem uma capacidade de transmissão constante, o tempo que o pacote leva para ser transmitido é o seu tamanho dividido pela capacidade do enlace. Isso impossibilita a modelagem precisa do sistema como uma fila M/M/1.

Para tornar mais claro o problema da dificuldade de modelagem, suponha o caso de um nó receber pacotes de apenas um emissor, os dois nós possuem a mesma capacidade de transmissão e os pacotes possuem todos o mesmo tamanho, como na figura C.1:

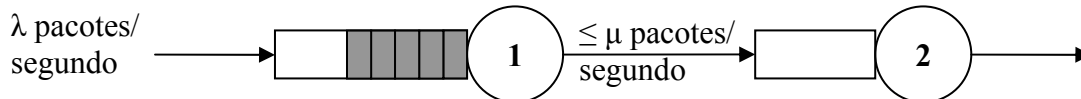


Figura C.1 – Dois nós com igual capacidade de transmissão. Se o tamanho dos pacotes é igual não há espera na fila do segundo nó.

No primeiro nó o processo de chegada dos pacotes é Poisson com taxa de chegada de pacotes por segundo λ , então essa é uma fila M/D/1 [10], pois como os pacotes tem tamanho igual se pode determinar o tempo gasto no atendimento do pacote. Na segunda fila, a taxa de chegada será no máximo μ , que é a taxa de atendimento do primeiro servidor e é igual a taxa de atendimento do segundo servidor. Assim quando o próximo pacote estiver preste a ser completamente montado pela camada de enlace da interface de recebimento o pacote anterior teve seus últimos bits transmitidos pela camada física da interface de envio. Dessa forma nunca haverá um pacote que necessite

esperar por atendimento no buffer do segundo nó e a modelagem dessa fila por M/M/1 traria resultados equivocados para as medidas de eficiência.

Supondo agora a mesma estrutura dos dois nós anteriores porém com o tamanho dos pacotes seguindo uma distribuição exponencial. Dessa forma a primeira fila pode ser modelada por M/M/1. Porém da mesma forma que no caso anterior a segunda fila não pode ser modelada por M/M/1, pois o tempo de chegada entre pacotes no segundo nó é fortemente correlacionado com o tamanho do pacote. Devido a isso, pacotes grandes irão esperar menos tempo na segunda fila do que na primeira, pois como o tempo gasto na sua transmissão é maior, o segundo nó tem mais tempo para atender os pacotes que estão na fila. Essa diferença do tempo entre chegadas pode ser observada na figura C.2:

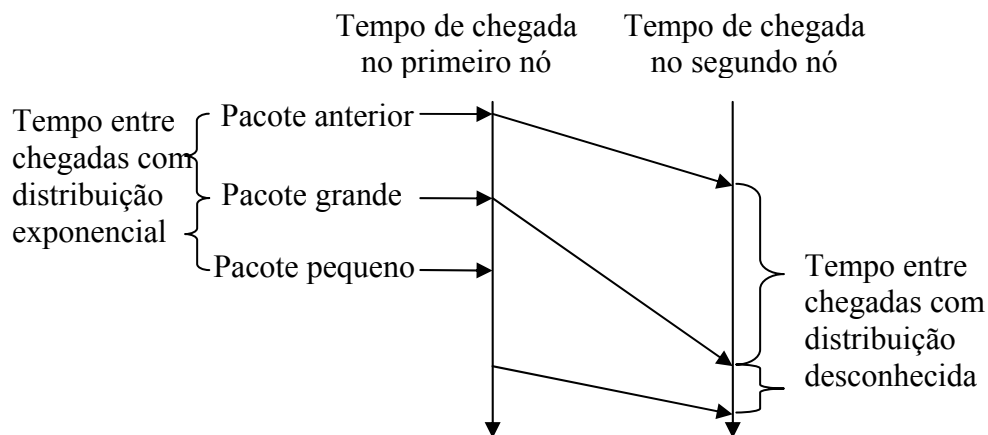


Figura C.2 – Diagrama dos tempos de chegada e do tempo gasto na transmissão dos pacotes.

Para que fosse possível uma aproximação do tempo de espera médio por pacote e do número de pacotes na rede de dados, Kleinrock sugeriu que a combinação de vários tráfegos de pacotes em um enlace de transmissão tem o efeito semelhante ao de restaurar a independência entre os tempos de chegada e os tamanhos dos pacotes. Essa combinação de tráfegos pode ser exemplificada na figura C.3:

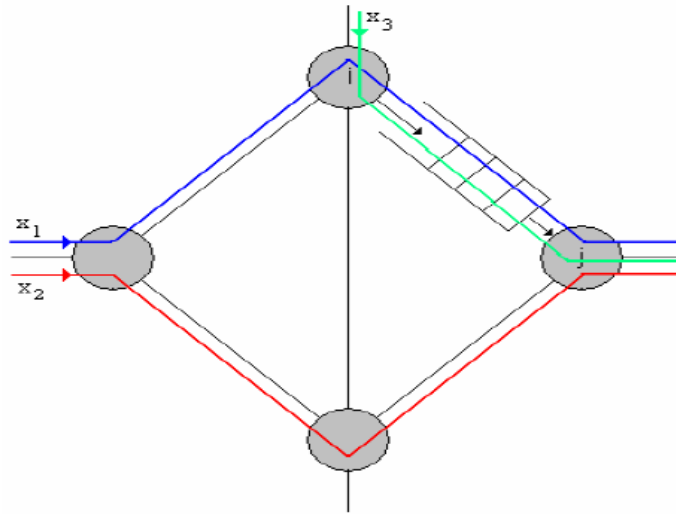


Figura C.3 – Exemplo de uma rede de dados.

Sendo X_n a taxa de chegada do tráfego n , a taxa de chegada no enlace (i,j) é a soma: $X_1 + X_3$. Assim $\lambda_{ij} = X_1 + X_3$.

Kleinrock concluiu assim que a fila de espera de cada um dos nós pode ser modelada por M/M/1 e parece ser uma aproximação razoavelmente boa para sistemas envolvendo tráfegos de chegada Poissonianos, tamanho dos pacotes exponencialmente distribuídos, uma rede densamente conectada e uma carga de tráfego moderada à alta. Assim podemos obter o número médio de pacotes no enlace (i,j) por :

$$N_{ij} = \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}}$$

Onde μ_{ij} é a taxa de atendimento de pacotes no enlace (i,j) e é menor ou igual à taxa de atendimento do nó j , será igual caso X_2 seja igual a zero e menor em caso contrário. E o tempo médio de espera por pacote no enlace (desprezando o tempo de processamento e propagação) é:

$$T_{ij} = \frac{1}{\mu_{ij} - \lambda_{ij}}$$

D Multiresolução

A multiresolução consiste na análise de sinais em diferentes escalas, a partir da projeção dos sinais em subespaços de $L^2(\mathcal{R})$ V_j e W_j . Os subespaços de escala V_j são definidos de forma que $V_0 \subset V_1 \subset V_2 \subset \dots$. Quanto maior o índice do subespaço, mais fino ele é, de forma que uma função que pertença a um subespaço V_j também pertence ao subespaço mais fino V_{j+1} . Uma função $f(t)$ possui projeções em cada subespaço V_j sendo que quanto mais fino for o subespaço, mais detalhes terá a projeção de $f(t)$ nesse subespaço, tendo a informação completa de $f(t)$ quando $j \rightarrow \infty$. Ou seja, designando a projeção de $f(t)$ em V_j por $f_j(t)$, temos que $f_j(t) \rightarrow f(t)$ quando $j \rightarrow \infty$. Como $V_j \subset V_{j+1}$ para todo j inteiro existe um espaço complementar W_j tal que $V_j \oplus W_j = V_{j+1}$. Ou seja, o espaço wavelet W_j contém os detalhes presentes em V_{j+1} e ausentes em V_j . Dessa forma, a função $\Delta f_j(t)$ dada por $\Delta f_j(t) = f_{j+1}(t) - f_j(t)$ pertence ao espaço W_j . Logo, temos que:

$$V_J = V_0 + \sum_{j=0}^{J-1} W_j \quad (D.1)$$

Para a análise de multiresolução temos basicamente 4 requerimentos fundamentais:

1. $V_j \subset V_{j+1}$, $\bigcap_j V_j = \{0\}$ e $\overline{\bigcup_j V_j} = L^2(\mathcal{R})$. A condição de esvaziamento

$\bigcap_j V_j = \{0\}$ quer dizer que $\|f(t)\| \rightarrow 0$ quando $j \rightarrow -\infty$. Portanto, a partir de

(1), vemos que V_J pode ser dado por: $V_J = \sum_{j=-\infty}^{J-1} W_j$. A condição de

completude $\overline{\bigcup_j V_j} = L^2(\mathcal{R})$ significa que $f_j(t) \rightarrow f(t)$ quando $j \rightarrow +\infty$, ou

seja $V_j \rightarrow L^2(\mathcal{R})$ quando $j \rightarrow \infty$.

2. O espaço V_{j+1} consiste das funções de V_j reescaladas, de forma que:

$$f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}.$$

3. Invariância a translação: Caso $f(t)$ pertença a V_j , todas as suas translações $f(t-k)$ também pertencem a V_j : $f(t) \in V_j \Leftrightarrow f(t-k) \in V_j$.
4. A base ortonormal de V_0 é formada por todas as translações inteiras da função de escalonamento $\phi(t)$.

As condições 2 e 3 podem ser generalizadas pela seguinte propriedade: $\phi(2^j t - k) \in V_j$ para qualquer j, k inteiros. A seguinte notação será adotada nesse trabalho: $\phi_{jk}(t) = 2^{j/2} \phi(2^j t - k)$. As funções $\phi_{jk}(t)$ são chamadas de funções de escalonamento. A partir das quatro condições anteriores pode ser provado que o espaço V_j possui base ortonormal completa dada por $\{\phi_{jk}(t)\}$.

Como $V_0 \subset V_1$, temos que todas as funções pertencentes ao espaço V_0 pertencem ao espaço V_1 e podem ser escritas como uma combinação linear das bases de V_1 . Logo a função $\phi(t)$, que pertence ao espaço V_0 pode ser escrita da seguinte forma:

$$\phi(t) = \sqrt{2} \sum_k c_k \phi(2t - k) \quad (D.2)$$

Em que:

$$c_k = \sqrt{2} \int_{-\infty}^{\infty} \phi(t) \phi(2t - k) dt \quad (D.3)$$

A equação (2) é chamada de equação de dilatação e pode ser generalizada da seguinte forma:

$$\phi_{jk}(t) = \sum_n c_n \phi_{j+1, 2k+n}(t) \quad (D.4)$$

Como já foi dito anteriormente, sempre existe um espaço W_j tal que $V_j \oplus W_j = V_{j+1}$. As ondeletas dadas por $\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k)$ são funções que pertencem ao espaço W_j .

Como $V_0 \oplus W_0 = V_1$, então $W_0 \subset V_1$. Logo, $\psi(t)$ pode ser escrito da seguinte forma:

$$\psi(t) = \sqrt{2} \sum_k d_k \phi(2t - k) \quad (D.5)$$

Em que:

$$d_k = \sqrt{2} \int_{-\infty}^{\infty} \psi(t) \phi(2t - k) dt \quad (D.6)$$

A equação (D.5) é chamada de equação da ondeleta e pode ser generalizada da seguinte forma:

$$\psi_{jk}(t) = \sum_n d_n \psi_{j+1, 2k+n}(t) \quad (D.7)$$

O subespaço \mathbf{W}_j possui base ortonormal $\{\psi_{jk}(t)\}$. É possível provar que as ondeletas são ortogonais às funções de escalonamento dentro da mesma escala, ou seja, $\psi_{jk}(t)$ e $\phi_{jm}(t)$ são ortogonais para quaisquer k e m . Além disso, as ondeletas são ortonormais em todas as escalas, ou seja, $\psi_{jk}(t)$ e $\psi_{JK}(t)$ são ortogonais para quaisquer $j \neq J$ e $k \neq K$ e possuem norma 1.

Como \mathbf{V}_{j+1} pode ser dado pela soma direta $\mathbf{V}_j \oplus \mathbf{W}_j$, uma função $f_{j+1}(t) \in \mathbf{V}_{j+1}$, representada em \mathbf{V}_{j+1} por $f_{j+1}(t) = \sum_k a_{j+1,k} \phi_{j+1,k}(t)$, pode ser representada em $\mathbf{V}_j \oplus \mathbf{W}_j$ por $f_{j+1}(t) = \sum_k a_{jk} \phi_{jk}(t) + \sum_k b_{jk} \psi_{jk}(t)$, em que os coeficientes são dados por:

$$a_{jk} = \int f_{j+1}(t) \phi_{jk}(t) dt \text{ e } b_{jk} = \int f_{j+1}(t) \psi_{jk}(t) dt$$

Da mesma forma, $f_{j+1}(t)$ pode ser representada em $V_{J_0} + \sum_{j=J_0}^J W_j$ por:

$$f_{j+1}(t) = \sum_k a_{J_0 k} \phi_{J_0 k}(t) + \sum_{j=J_0}^J \sum_k b_{jk} \psi_{jk}(t) \quad (D.8)$$

Quando temos $f(t) \in L^2(\mathcal{R})$, chegamos à seguinte representação de um sinal real:

$$f(t) = \sum_k a_k \phi_{J_0 k}(t) + \sum_{j=J_0}^{\infty} \sum_k b_{jk} \psi_{jk}(t) \quad (D.9)$$

Assim, o sinal pode ser representado no domínio das wavelets através dos coeficientes a_{jk} e b_{jk} , a partir das funções de base estabelecidas $\psi(t)$ e $\phi(t)$. Essa representação define a Transformada Discreta de Wavelet, cuja transformada inversa é dada por (D.9).

Os coeficientes a_{jk} e b_{jk} relativos à projeção da função $f(t)$ em V_j e W_j , respectivamente, podem ser obtidos a partir dos coeficientes $a_{j+1,k}$ relativos a projeção de $f(t)$ em V_{j+1} da seguinte forma:

$$a_{jk} = \sum_{\ell} c_{\ell-2k} a_{j+1,\ell} \quad (D.10)$$

$$b_{jk} = \sum_{\ell} d_{\ell-2k} a_{j+1,\ell} \quad (D.11)$$

Além disso, podemos obter os coeficientes $a_{j+1,k}$ a partir de a_{jk} e b_{jk} utilizando a seguinte equação:

$$a_{j+1,n} = \sum_k c_{n-2k} a_{jk} + \sum_k d_{n-2k} b_{jk} \quad (D.12)$$

D.1 Wavelets de Haar

Uma das famílias de ondeletas mais conhecidas são as chamadas wavelets de Haar. Para o caso das wavelets de Haar, a equação de dilatação é dada por (D.2) com $c_0 = c_1 = \frac{1}{\sqrt{2}}$ e $c_k = 0$ para $k \neq 0$ e $k \neq 1$. Dessa forma, a partir da equação de dilatação obtemos:

$$\phi(t) = \phi(2t) + \phi(2t-1) \quad (D.13)$$

Cuja solução é:

$$\phi(t) = \begin{cases} 1, & 0 \leq t < 1 \\ 0, & \text{c.c.} \end{cases} \quad (D.14)$$

Ilustramos abaixo o gráfico de $\phi(t)$ e os gráficos de $\phi_{jk}(t)$ para $j = 0, 1$ e 2 e $k = 0, 1, 2, 3$ e 4 .

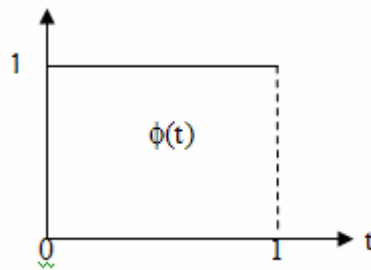


Figura D.1 – Gráfico de $\phi(t)$

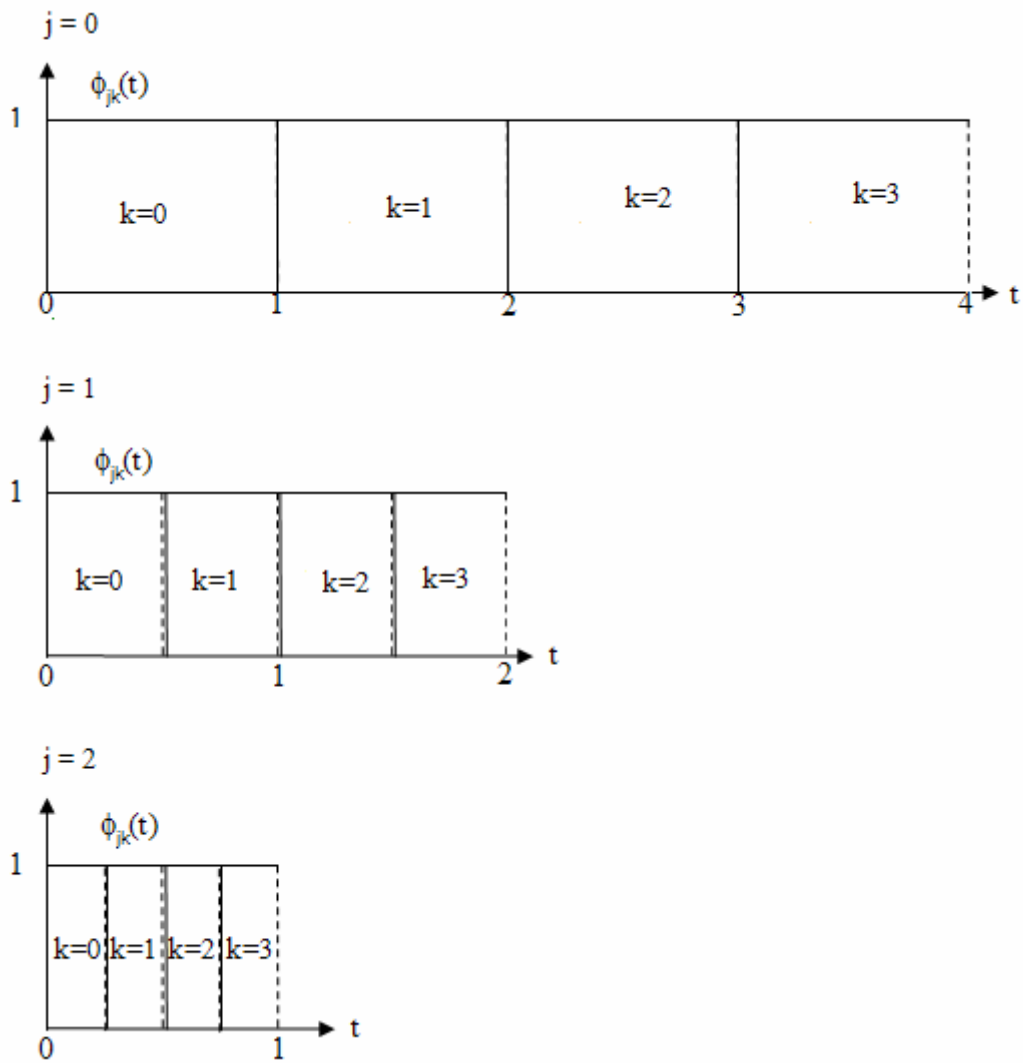


Figura D.2 – Gráficos de $\phi_{jk}(t)$

Podemos perceber pelas figuras D.1 e D.2 que, como era esperado, as funções $\phi_{jk}(t)$ são ortonormais dentro da mesma escala j . Além disso, podemos perceber que qualquer função de V_j , construída a partir de suas funções de base $\{\phi_{jk}(t)\}$, podem ser construídas a partir das funções de base de um espaço mais fino $V_m\{\phi_{mk}(t)\}$, em que $m > j$.

A equação da ondeleta das wavelets de Haar é dada por (D.7) com $d_0 = -d_1 = \frac{1}{\sqrt{2}}$ e $d_k = 0$ para $k \neq 0$ e $k \neq 1$. Dessa forma, a partir da equação de ondeleta temos:

$$\psi(t) = \phi(2t) - \phi(2t - 1) \quad (D.15)$$

As figuras D.3 e D.4 mostram a ondeleta de Haar $\Psi(t)$ e os gráficos de $\Psi_{jk}(t)$ para $j = 0$ e 1 e $k = 0$ e 1 .

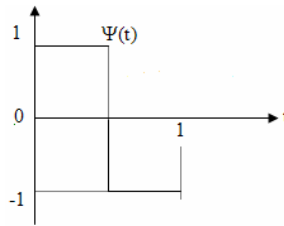


Figura D.3 – Wavelet de Haar

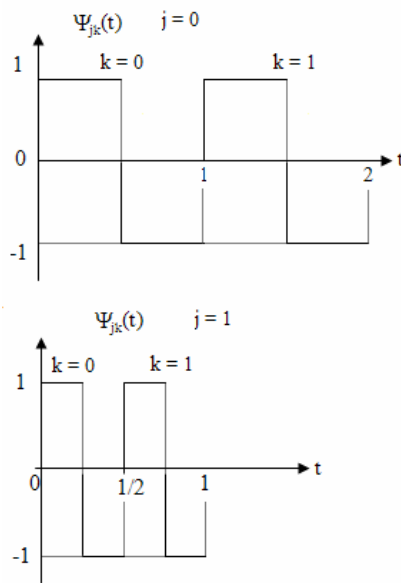


Figura D.4 – Gráficos de $\Psi_{jk}(t)$

Pelas figuras D.3 e D.4 podemos perceber que as wavelets de Haar $\Psi_{jk}(t)$ são ortogonais às funções de escalonamento $\phi_{jk}(t)$ dentro da mesma escala j . Percebemos também que as wavelets de Haar são ortonormais em todas as escalas, isto é, $\Psi_{jk}(t)$ possui norma 1 e $\int_{-\infty}^{+\infty} \Psi_{jk}(t)\Psi_{JK}(t)dt = 0$ para $j \neq J$ e/ou $k \neq K$. Além disso, vemos que é possível obter qualquer função de base de V_{j+1} a partir de uma combinação linear das funções de base de V_j e W_j . Logo, qualquer função pertencente a V_{j+1} pode ser escrita utilizando-se as funções de base de V_j e W_j .

De acordo com (D.10) e (D.11), temos que os coeficientes a_{jk} e b_{jk} , relativos à projeção da função $f(t)$ em V_j e W_j , podem ser obtidos a partir dos coeficientes $a_{j+1,k}$ da seguinte forma:

$$a_{j,k} = \frac{1}{\sqrt{2}}(a_{j+1,2k} + a_{j+1,2k+1}) \quad (D.16)$$

$$b_{j,k} = \frac{1}{\sqrt{2}}(a_{j+1,2k} - a_{j+1,2k+1}) \quad (D.17)$$

Além disso, de acordo com a equação (D.12), percebemos que os coeficientes $a_{j+1,k}$ são obtidos a partir de a_{jk} e b_{jk} pelas seguintes equações:

$$a_{j+1,2k} = \frac{1}{\sqrt{2}}(a_{jk} + b_{jk}) \quad (D.18)$$

$$a_{j+1,2k+1} = \frac{1}{\sqrt{2}}(a_{jk} - b_{jk}) \quad (D.19)$$

Como as funções de escala de Haar $\phi_{jk}(t)$ são funções retangulares, os coeficientes de escala a_{jk} representam a média local do sinal no intervalo $[k2^{-j}, (k+1)2^{-j}]$. Logo os coeficientes a_{jk} 's representam uma aproximação em tempo discreto do sinal na escala de resolução j .

Referências Bibliográficas

- [1] Leland, W. et. al. On the self-similar nature of ethernet traffic (extended version), **IEEE/ACM Trans. on Networking**, vol. 2, no. 1, pp. 1--15, 1994.
- [2] Taqqu, M. S., Willinger, W. e Sherman, B. Proof of a fundamental result in self-similar traffic modeling, **Computer Communication Review**, vol. 27, p. 5-23, 1997.
- [3] Huang, J. Generalizing 4IPP Traffic Model for IEEE 802.16.3, **IEEE 802.16.3c-00/58**, Meeting #11, Ottawa, Dezembro 2000.
- [4] Ledesma, S. e Liu, D. A Fast Method for Generating Self-Similar Network Traffic, **Proceedings of the 2000 International Conference on Communication Technologies**, Beijing, China, p.54-61, Agosto 2000.
- [5] Paxson, V. Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic, **Computer Communication Review**, vol.27, p.5-18.
- [6] Silva, P. R. T. **Estudo do Tráfego Auto-similar em Redes Multiserviço**, (Projeto Final de Graduação), Departamento de Engenharia Elétrica, UnB, 2005.
- [7] Bizerra, R. S. **GTAR – Uma Ferramenta para Análise de Desempenho em Redes Convergentes**, (Projeto Final de Graduação), Departamento de Engenharia Elétrica, UnB, 2006.
- [8] Castro e Silva, J. L. **ProCon - Prognóstico de Congestionamento de Redes de Computadores usando Wavelets**, (Tese de Doutorado), Universidade Federal de Pernambuco, 2004.
- [9] Law, A.M. e Kelton, W. D. **Simulation Modeling and Analysis**, 2nd ed. Nova Iorque: McGraw-Hill, 1991.
- [10] Gross, D. e Harris, C. M. **Fundamentals of Queueing Theory**, 3rd ed. Nova Iorque: John Wiley & Sons, 1998.
- [11] Kettani, H. e Gubner, J.A. A Novel Approach the Estimation of the Hurst Parameter in Self-similar Traffic, **Proceedings of IEEE Conference on Local Computer Networks**, Tampa, Flórida, Novembro 2002.
- [12] Karagiannis, T., Faloutsos, M. e Riedi, R.H. Long-Range Dependence: Now You See It, Now You Don't!, **Proc. IEEE Global Telecommunications Conf. Global Internet Symposium**, 2002.
- [13] Clegg, R. A Practical Guide to Measuring the Hurst Parameter, **Proceedings of 21st UK Performance Engineering Workshop**, School of Computing Science, 2005
- [14] Andersen, A. T. e Newlsen, B. F. A Markovian Approach for Modeling Packet Traffic with Long-Range Dependence, **IEEE Journal on Selected Areas in Communications**, v.16, n.5, p.719-732, Junho 1998.

- [15] Gubner, J. A. Theorems and Fallacies in the Theory of Long-Range-Dependent Processes, **IEEE Transactions on Information Theory**, v.51, n.3, p.1234-1239, Março 2005.
- [16] Riedi, R. An introduction to multifractals, **Relatório Técnico**, Rice University, 1999. Disponível em: < <http://www.stat.rice.edu/~riedi/> >. Acesso em: julho 2006.
- [17] Abry, P. et. al., The multiscale nature of network traffic: Discovery, Analysis, and Modelling, **IEEE Signal Processing Magazine**, vol. 19, no. 3, pp. 28-46, Maio 2002.
- [18] Song, S., Ng, J. K. e Tang, B. Some Results on the Self-similarity Property in Communication Networks, **IEEE Transactions on Communications**, v. 52, n.10, p.1636 - 1642, Outubro 2004.
- [19] Ribeiro, V. J. et. al. Simulation of non-gaussian long-range-dependent traffic using wavelets. **Proceedings of ACM SIGMETRICS '99**, p. 1-12, Maio 1999.
- [20] Ribeiro, V.J., Reidi, R.H. e Baraniuk, R.G. Wavelets and Multifractals for Network Traffic Modeling and Inference, **ICASSP 2001**.