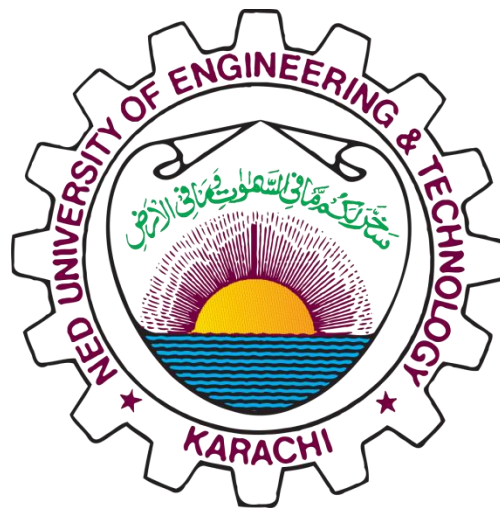


**INTERNET TRAFFIC ANALYSIS OF ACCESS AND CORE NETWORKS USING  
SELF-SIMILAR PROCESSES**

**INDEPENDENT STUDY PROJECT REPORT**

**By**

**Sabra Ayub**



**Department of Computer and Information Systems Engineering**

**NED University of Engineering & Technology**

**Karachi-75270**

**INTERNET TRAFFIC ANALYSIS OF ACCESS AND CORE NETWORKS USING  
SELF-SIMILAR PROCESSES**

**INDEPENDENT STUDY PROJECT REPORT**

**By**

**Sabra Ayub**

**CS-077/20117-18**

**Supervisor**

**Dr. Asad Arfeen**

**July 2018 - April 2019**

## ABSTRACT

With the emergence of twenty-first century, high-speed networks have been come into light, and internet has achieved the status of the most complex network humans have ever invented. Therefore, its dynamics become labyrinthine and traffic growth expeditious. Further, the prior models to understand the nature of internet traffic such as Markovian simulated models rendered unfit to explain the auto-correlation and burstiness of traffic. As time advances, the studies for Failure of Poisson models and discovery of Self- Similarity in both- Access and Core networks materialized. Further, the traditional models for internet traffic modeling on data characterized by self-similarity yielded incorrect observations for networks parameters hence lead to overestimation of network resources and inefficient allocation of resources. Parsimony of internet traffic models, also become necessary to meet the needs of Future Internet. In this ISP, the Internet traffic analysis of Access and Core networks using Self-Similar processes, the methods for estimation of self-similarity parameter  $H$ , such as Wavelet Transformed, Rescaled Ranged Transform, Variance - Time Plot and Periodogram, are first evaluated and the most efficient one is selected for furthering the study. Along with it , an internet traffic repository has created for both networks, which is based upon its behavior of long-range dependence at coarse and fine scales, to meet the need network performance optimization. Moreover, the network traffic behaviors are also thoroughly studied to find the correlation among it. Multiscaling behavior of traffic is also examined using Wavelet transformed. Afterwards, the role of Application layers data traffic in the burstiness is studied and results are formulated. Finally, applications of such work are then demonstrated to mark the anomalous behavior of network traffic for prevention of device failures, congestion of networks and malicious intrusions.

## **ACKNOWLEDGEMENT**

Many people contributed in the completion of this ISP report in different ways. I would like to acknowledge in particular Dr.Asad Arfeen for instilling me confidence to work in such domain, of which I have little knowledge. All that I have done so far is under his supervision and assistance, despite his plethora of responsibilities. I also want to express my special thanks to Sir Umer Mukhtar who has helped me in data archiving phase of this project and Miss Sumayya Zafar for various discussion on the estimators of self-similarity.

## TABLE OF CONTENTS

<b>LIST OF FIGURES</b> .....	<b>8</b>
<b>LIST OF TABLES</b> .....	<b>10</b>
<b>CHAPTER 1 INTRODUCTION</b> .....	<b>11</b>
1.1 OVERVIEW .....	11
1.2 WHY SELF-SIMILAR PROCESSES ARE IMPORTANT? .....	11
1.3 OBJECTIVES OF THE ISP .....	12
1.4 CONTRIBUTION OF THE ISP .....	12
1.5 DESCRIPTION OF THE TRAFFIC TRACES .....	12
1.6 METHODOLOGY FOR ANALYSIS .....	13
1.7 ORGANIZATION OF THE ISP .....	14
1.8 SUMMARY OF THE CHAPTER .....	14
<b>CHAPTER 2 SELF SIMILAR PROCESSES</b> .....	<b>16</b>
2.1 INTRODUCTION .....	16
2.1.1 SELF-SIMILARITY .....	16
2.1.2 CHAOS AND RANDOMNESS .....	17
2.1.3 FRACTALS .....	18
2.2 TRAFFIC SELF-SIMILARITY .....	19
2.2.1 STATIONARY RANDOM PROCESSES .....	19
2.2.2 SELF-SIMILAR PROCESSES .....	20
2.3 PROPERTIES OF SELF-SIMILAR PROCESS .....	22
2.3.1 SLOWLY DECAYING VARIANCE .....	22
2.3.2 HURST EFFECT .....	23
2.3.3 1/F NOISE .....	24
2.3.4 HEAVY- TAILED DISTRIBUTIONS .....	25
2.4 SELF-SIMILAR TRAFFIC MODELS .....	26
2.4.1 AGGREGATE TRAFFIC MODELS .....	26
2.4.1.1 FRACTIONAL BROWNIAN MOTION .....	26
2.4.1.2 FRACTIONAL ARIMA MODELS .....	27
2.4.2 SINGLE SOURCE MODEL .....	28
2.4.2.1 PARETO DISTRIBUTION .....	28
2.4.2.2 GENERALIZED PARETO DISTRIBUTION .....	28

2.5 SUMMARY OF THE CHAPTER.....	29
<b>CHAPTER 3 HURST PARAMETERS ESTIMATION TECHNIQUES.....</b>	<b>30</b>
3.1 INTRODUCTION .....	30
3.2 METHODS OF ANALYZING SELF-SIMILARITY .....	30
3.2.1 R/S TRANSFORM .....	30
3.2.2 WAVELET BASED H ESTIMATOR.....	31
3.2.3 PERIODOGRAM BASED ESTIMATORS .....	33
3.2.4 VARIANCE TIME METHOD .....	33
3.3 MEAN SQUARE ERROR AND R/S TRANSFORM: .....	34
3.4 MEAN SQUARE ERROR AND WAVELET TRANSFORM METHOD: .....	36
3.5 MEAN SQUARE ERROR AND PERIODOGRAM.....	38
3.6 MEAN SQUARE ERROR AND VARIANCE TIME.....	40
3.7 SUMMARY OF THE CHAPTER.....	42
<b>CHAPTER 4 INTERNET TRAFFIC ARCHIVE.....</b>	<b>43</b>
4.1 INTRODUCTION .....	43
4.2 EXISTING INTERNET TRAFFIC ARCHIVES .....	45
4.2.1 THE CENTER FOR APPLIED INTERNET DATA ANALYSIS (CAIDA).....	45
4.2.2 WAIKATO INTERNET TRAFFIC STORAGE (WITS).....	45
4.2.3 WIDE PROJECT (MAWI GROUP).....	45
4.2.4 INTERNET TRAFFIC STATISTICS ARCHIVE (ITSA): .....	46
4.3 METHODS OF INTERNET TRAFFIC ARCHIVING .....	46
4.3.1 FULL PACKET CAPTURE .....	47
4.3.2 NETWORK FLOW CAPTURE .....	48
4.3.3 AUGMENTED FLOW CAPTURE.....	49
4.4 INDEXING OF TRACES.....	50
4.4.1 BITMAP INDEXING .....	50
4.4.1.1 COMPRESSION.....	51
4.4.1.2 ENCODING.....	51
4.4.2 H- BASED INDEXING.....	51
4.5 NRPU – INTERNET TRAFFIC ARCHIVE .....	52
4.5.1 DESCRIPTION OF TRACES .....	53
4.5.2 GRAPHICAL PRESENTATION OF TRACES .....	53

4.5.2.1 PROTOCOLS DISTRIBUTION OF A TRACE FILE.....	54
4.5.2.2 PERCENTAGE OF PACKETS PER PROTOCOL.....	54
4.5.2.3 PACKET COUNT IN A TRACE FILE.....	55
4.6 SUMMARY OF THE CHAPTER.....	56
<b>CHAPTER 5 APPLICATION LAYER TRAFFIC AND SELF-SIMILARITY .....</b>	<b>57</b>
5.1 INTRODUCTION .....	57
5.2 INTERNET TRAFFIC PROTOCOL BREAKDOWN.....	57
5.3 APPLICATION LAYER DEEP PACKET INSPECTION .....	59
5.4 AGGREGATED TRAFFIC & SELF-SIMILARITY .....	59
5.5 ANALYSIS OF WHATSAPP VOICE, FACEBOOK, GOOGLE TRAFFIC.....	60
5.5.1 SELF - SIMILARITY INDUCED BY DIFFERENT BY HTTPS PROTOCOLS .....	60
5.5.1.1 WHATSAPP & SELF-SIMILARITY.....	60
5.5.1.2 WHATSAPP VOICE AND SELF-SIMILARITY.....	61
5.5.1.3 FACEBOOK AND SELF-SIMILARITY.....	61
5.5.1.4 GOOGLE AND SELF-SIMILARITY.....	62
5.5.2 SELF-SIMILARITY INDUCED BY INDIVIDUAL PROTOCOLS.....	63
5.5.2.1 TRAFFIC WITHOUT WHATSAPP & SELF-SIMILARITY.....	62
5.5.2.2 TRAFFIC WITHOUT WHATSAPP VOICE AND SELF-SIMILARITY.....	63
5.5.2.3 TRAFFIC WITHOUT FACEBOOK AND SELF-SIMILARITY.....	64
5.5.2.4 TRAFFIC WITHOUT GOOGLE AND SELF-SIMILARITY.....	65
5.6 RESULTS .....	65
5.7 INTERNET TRAFFIC ANOMALY DETECTION BASED ON SELF-SIMILARITY.....	66
5.8 SUMMARY OF THE CHAPTER.....	67
<b>CHAPTER 6 CONCLUSION AND FUTURE WORK.....</b>	<b>68</b>
<b>REFERENCES.....</b>	<b>69</b>

## LIST OF FIGURES

Figure 2.1: Lorenz Attractor When $p = 28$ , $\Sigma = 10$ And $b = 8/3$ Resembling a Butterfly, Showing How the Chaotic System Forms a Deterministic Shape.....	17
Figure 2.2: Koch Curve with Four Levels of Iteration with Reference Shape of Equilateral Triangle. ....	18
Figure 2.3: Traffic Trace on Tour Different Time Scales Depicting Self-Similar Characteristics .....	20
Figure 2.4: Fractional Brownian Motion with $H = 0.7, 0.8, 0.9$ and $n = 100$ .....	21
Figure 2.5: Variance Time Plot Showing the Decaying Constant $\beta = -0.152$ Having Negative Slope and $H = 0.924$ .....	22
Figure 2.6: R/S statistic Plot Showing $H = 0.84$ for a Traffic Trace.....	24
Figure 2.7: $1/f$ Noise of Traffic Trace Showing the Basic Characteristic of Apparently Random Fluctuations but Follow a Self-Similar Pattern.....	25
Figure 2.8: Probability Density Function of Pareto Distribution .....	26
Figure 2.9: Fractional Brownian Motion .....	27
Figure 2.10: Autocorrelation Function of Time Series X at various lags .....	27
Figure 3.1: R/S Plot of a Traffic Trace along with a Slope Depicting $H = 0.957$ .....	31
Figure 3.2: Wavelet Estimator: Energy vs. Octave j Plot to Estimate H by Slope of Regression Line; $H = 0.994$ .....	32
Figure 3.3: Variance Time log-log Plot for a Traffic Trace; the Regression Line Giving the Value of $H = 0.834$ .....	34
Figure 3.4: Estimated Value of $H'$ by R/S Transform and Corrected Value of H after Error Calculation by MSE of 0.19 Per cent.....	36
Figure 3.5: Estimated Value of $H'$ by Wavelet Transform and Corrected Value of H after Error Calculation by MSE of 0.025 Percent.....	38
Figure 3.6: Estimated Value of $h'$ by Periodogram and Corrected Value of h After Error calculation by MSE of 0.20 Percent.....	40
Figure 3.7: Estimated Value of $h'$ by Variance Time Method and Corrected Value of h after error calculation by MSE of 0.20 Percent.....	42
Figure 4.1: Growth of Internet Users in Different Countries from 1960s to 2017 .....	43
Figure 4.2: Steps of Advancements in the Internet Networks – From Analog World to Digital Universe.....	44
Figure 4.3: The component of ITSA .....	46
Figure 4.4: Information of Packet Captured by Wireshark.....	47
Figure 4.5: A Hurst Parameter Based Index of Trace Files .....	52



Figure 4.6: A Trace Files along with the summary of Trace .....	53
Figure 4.7: Bar Chart - Protocol Distribution of a Trace .....	54
Figure 4.8: Percentages of Packets for each protocol. ....	55
Figure 4.9: Packet Count of a Trace File .....	56
Figure 5.1: Percentages of Packets for each Protocol. ....	57
Figure 5.2: Experimental Setup for Traffic Trace Capture .....	58
Figure 5.3: A Traffic Trace Capture by Deep Packet Inspection.....	59
Figure 5.4: Measure of Self-Similarity $h= 0.989$ by Wavelet Scaling Analysis Method for Aggregated Traffic Trace of 15 minutes Composed of 132,099 Packets .....	59
Figure 5.5: Measure of Self-Similarity $H= 0.786$ by Wavelet Transform for WhatsApp packets for 800. 60	
Figure 5.6: Measure of Self-similarity $h= 0.823, 0.583$ by V-T and Periodogram for WhatsApp packets for 1200.....	61
Figure 5.7: Measure of Self-similarity $h= 0.934, 0.829$ by Wavelet and V-T Method.....	62
Figure 5.8: Measure of Self-similarity $h= 0.9967$ Periodogram for SSL.Google.....	62
Figure 5.9: Measure of Self-similarity $h= 0.841$ by R/S Transform without WhatsApp.....	63
Figure 5.10: Measure of Self-similarity $h= 0.781$ R/S Transform for without WhatsAppVoice .....	64
Figure 5.11: Measure of Self-similarity $h= 0.841$ by R/S transform without Facebook.....	64
Figure 5.12: Measure of Self-similarity $h= 0.843$ by R/S transform without Google .....	65
Figure 5.13: Manifesting the Value of Hurst parameter with https Applications and without them; a difference of 15 percent to 21 percent in value of H is observed.....	65
Figure 5.14: Manifesting the Value of Hurst parameter which is 1.443 out of range of $(0.5 < h < 1)$ showing the anomalous behaviour of traffic. ....	66

## LIST OF TABLES

Table 1.1. Dataset of Captured Network Traces .....	13
Table 3.1. Estimated and Corrected Value of H by R/S Transform after MSE Analysis .....	35
Table 3.2. Estimated and Corrected Value of H by Wavelet Transform after MSE Analysis.....	37
Table 3.3. Estimated and Corrected Value of H by Periodogram after MSE Analysis .....	39
Table 3.4. Estimated and Corrected Value of H by Variance Time Method .....	41
Table 4.1. Comparison of Unidirectional and Bidirectional Flow .....	49
Table 4.2. Types of Capturing Network Traces for Internet Traffic Archiving.....	50

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 OVERVIEW**

Internet traffic continues to exhibit rapid traffic growth. One of the leading sources monitoring Internet traffic levels indicates that the growth is 40 to 50 percent annually [1]. The future of internet technology will demand enough expertise for the daunting challenges of its novel usage. This is because of its growing applications in all fields. Due to future implications of internet technology this area has attracted many research and development programs to understand the current infrastructure and make it resilient for future. This project is an initiative to be the part of development of Future Internet (FI). The current project is a subset of larger project: Stochastic Modeling and Real-time Estimation of Emerging Internet Traffic in Access and Core Networks. This project aims to develop a repository of internet traffic based on the Self-Similarity Parameter H, Hurst estimator, to help in studies of marking the anomalous behavior of network traffic and anticipate in robust management of network devices against failures, congestion and malicious intrusions [2].

Self-similarity in traffic engineering is considered as one of the important characteristic of network traffic. As Paxson [3] found that the conventional models of circuit switching networks are inefficient to describe the nature of fast-networks; he proposed the possible self-similarity in them. The notion of self-similarity was coined by Kolmogorov which later on brought in to light by Mandelbrot [4].

#### **1.2 WHY SELF-SIMILAR PROCESSES ARE IMPORTANT?**

Stochastic processes that have the characteristic self-similar occurrence are called self-similar process. This area become an important arena of analysis and modeling in traffic engineering, network performance and network dimensioning ; after formative works of Leland, Taqqu ,Willinger, Crovella and et all [5, 6, 7] . The question in hand is why self- similarity is gained importance. The answer lays in the findings of Paxson that conventional Poisson model failed to fully define the characteristic of Wide area network traffic. However, the traffic according to some studies [8], in Access network is resembled to Long Range Dependent (LRD) and self-similar contrary to Short Range Dependent (SRD) in Core networks. But self- similarity plays its important role in queuing, network performance, resources management, congestion control and anomalous behavior detection. So, importance of consideration of self-similarity of internet traffic in Access and Core networks are listed below briefly.

- Self-similar processes helps in designing of traffic models in a parsimonious manner.
- Self-similar traffic manifests the persistence of clustering and burstiness which degrades network performance, however Poisson modeled traffic does otherwise, so for resilient network it must be analyzed.
- Network Quality of Services (QoS) rest on how to deal with the rare events such as traffic peaks (bursts) that could induce network failures because of packets losses, queue overflow and delay bounds.
- Self-similar models fit Ethernet data better than traditional traffic models, as they ignore the presence of long-term correlation of data [9].

### **1.3 OBJECTIVES OF THE ISP**

The objectives of this ISP are as follows:

- Hurst parameter based indexing of internet traffic and its archive
- A model of Internet traffic of internet traffic for access and core network capable of zooming in and out of various time scales
- Time series decomposition of Internet traffic from higher to lower time scales.

### **1.4 CONTRIBUTION OF THE ISP**

This ISP proposed an analytical technique for understanding the nature of network traffic in order to properly design and implement computer networks and network services like the World Wide Web. As internet network saturation is touching its boundaries due to growth of various factors such as real time applications, games, social networks high-speed network links. This project aims to provide a platform to understand the dynamics of current internet traffic by various statistical approaches based on self-similarity of traffic data and also manifested the Multiscaling nature of internet traffic. As network traffic usually depict self-similar behavior but this property lost during anomalous conditions such as device failure, congestion and malicious intrusions. Therefore, this project also contributed in highlighting such conditions. The application layer traffic and its contribution in inducing self-similarity is analyzed in this project to better devise network strategies with regard to ubiquitous computing and future internet (FI).

### **1.5 DESCRIPTION OF THE TRAFFIC TRACES**

One of the main components of any research project is its dataset. The data for this project is captured through a traffic capturing software, Wireshark, locally to understand the pattern of trace in the local

networks, their dynamics and how are they using resources thoroughly. The packet traces are of PCAP format afterwards processed in R language for drawing conclusions.

**Table 1.1. Dataset of Captured Network Traces**

Measurement Period				Total number of bytes	Total number of packets
August 25 <sup>th</sup> ,2018	3:00 - 3:15	Busy hour		78152230000	138438635 (78152.23MB)
	Pm				
August 26 <sup>th</sup> ,2018	3:00 - 3:15			111898370000	127530382 (111898.37MB)
	Pm				
September 2 <sup>nd</sup> ,2018	9:00 - 9:30	Low hour		52430000	12796 (52.43MB)
	Am				
September 3 <sup>rd</sup> ,2018	9:00 -9:30			64120000	20,960 (64.1MB)
	Am				
November 10 <sup>th</sup> ,2018	12:45 -1:00	Normal hour		1762530000	8438736 (1762.53MB)
	Pm				
December 30 <sup>th</sup> ,2018	2:00 2:15			2066870000	9029805 (2066.87MB)
	Pm				

In this ISP, the network traces are collected in busy hours, low hours and normal hours for the duration of 15 to 30 minutes, description of data set is in Table 1.1. The traces are collected by Wireshark from access and core networks .After capturing of packets they are anonymized to ensure the security. The technique for anonymization of IP packets are one on one mapping. The traces are of in the range of few MBs to hundreds and thousands of MBs.

## 1.6 METHODOLOGY FOR ANALYSIS

The structural study of internet traffic is based upon packets, flows and sessions. In this project the focus is based upon packets; recent studies have shown that the presence of significant statistical properties in packet traffic that are more characteristic of fractal processes than conventional stochastic processes [10]. Hence, self-similarity, heavy tailed distribution, long-range dependence or slowly decaying autocorrelations; and fractal dimensions, all of which have been observed in actual packet traffic traces are the covered in this project. A packet has two parts: control information and user data-payload. Control information monitors the delivery of payload. It has source and destination network addresses, error detection codes, and sequencing information. This is concatenated with packet headers and trailers. There

is variety of tools for packet capturing purpose as listed in [11]. After capturing of packets they are processed to anonymized. And Count Data of trace files are obtained using tcpdump library<sup>1</sup>.the count data is than undergone though process of finding the underlying pattern in them and their characteristics. Such as methods of finding self-similarity in them: Rescaled Range method [12], Variance Time plot [13], Wavelet Transform [14], and Periodogram [15]. Then the result would be analyzed to reach at the particular conclusions for practical applications.

### **1.7 ORGANIZATION OF THE ISP**

Chapter one would be discussing the overview of the project and the importance of self-similar processes to be considered for the study. Objectives, contributions and description of traffic trace are also listed in this chapter. Methodology that would be perused in the thesis would be discussed. Further, chapter two is about Self-Similar processes, its properties and traffic models followed by such traffic. Chapter three includes methods of Analyzing Self-similarity, mean Square Error for R/S Transform, Wavelet Transform, Periodogram and Variance Time methods. Internet traffic archive is discussed in detail after a brief introduction of existing Internet Traffic Archives and methods of Internet Traffic Archiving. The techniques for indexing are also part of this chapter. Application layer traffic and self-similarity is the fifth chapter of the project which discusses the practical usage of techniques for finding self-similarity, an analysis of https application contribution in the self-similarity of aggregated traffic would than demonstrated. An Image of protocol break down in typical traffic trace would be than explained and results would be concluded. Last chapter of the project would be discussed the future dimension of the domain of self-similarity and its practical applications.

### **1.8 SUMMARY OF THE CHAPTER**

The contemporary internet infrastructure is proceeding towards its permissible limits [1]. As it is a measure component of today's world for information and communication to growth and development, its future must continue to foster and support its novel usage. In order to address the needs of Future Internet (FI), a number of initiatives have been taken worldwide to understand the underlying dynamics of present day internet for the sustainable development of future internet. These Initiatives include Future Internet (FI) Forums in Europe, USA and South Korea. They are providing platforms for Research and Development (RND) to test protocols of future Internet design and related performance evaluation studies. The purpose of this project is inspired by these initiatives to stand in line with the world in the

---

<sup>1</sup> [www.tcpdump.org](http://www.tcpdump.org)

development of Future Internet (FI). In order to fulfill this objective an internet traffic repository will be developed and maintained. The striking feature of this repository is its classification based on the Self-Similarity Measurement using Hurst Parameter to mark the anomalous behavior of network traffic and help it against device failure, congestion and malicious intrusions [2]. This aims to determine a structural model of Internet teletraffic for both access and backbone core networks. This will be used for performance evaluation and optimization of various information services of Future Internet (FI), allowing efficient sharing of resources and increasing network security as well as performance of the network.

## **CHAPTER 2**

### **SELF SIMILAR PROCESSES**

#### **2.1 INTRODUCTION**

Statistical analysis of traffic measurements from a various packet networks as discussed in [3, 5, 6, 13, 12, 13] such as Ethernet, CCSN/SS7, ISDN and VBR video over ATM, have demonstrated the fractal-like or Long Range Dependent (LRD) behavior. Such behavior cannot be captured by traditional telephony models [3] as the aggregating stream intensifies the burstiness rather than smoothing it out as happened in Poisson modeled Markovian processes and others. The fractal-like behavior of traffic from packet switch networks has wide impact on network performance, resources management, and design of networks. Apart from LRD, some studies have also shown that internet traffic also behaves as Short Range Dependent (SRD) and phenomenon of superposition is very common as traffic flows from access to core networks [17,19]. The question arises here is which behavior of traffic is relevant for practical purposes. Various studies have thrown their weight behind LRD because of traffic measurements, queuing behavior and buffer sizing [18, 19, 20], admission control [21] and congestion control [23] support it. Further, the traditional models show exponential tail behavior and thus results in overly optimistic measures for queuing and performance of network. Whereas, traffic behaviors is hyperbolic decay or Weibull tail decay [21] and impact the design , control, buffer size and Quality of Service (QoS) provisions of networks. Interestingly, such processes have also been observed in other areas like: hydrology, biophysics, financial economics [22], and the work on fractal dynamics of network traffic behavior has been going on for only two decades. Self-similarity means a particular correlation structure is retained over a wide range of time scales, whether geographically or mathematically. Hence, for the purpose of directing the course of such processes, one must validate the possible reasons responsible for generating self-similarity in network traffic.

#### **2.1.1 SELF-SIMILARITY**

Self-similarity means scale invariance, that is, a process displaying structural similarities across a wide range of scales. Simply put that the reference structure is repeating itself over a wide range of scales of diverse dimensions, which could be either geometrically or statistically, or temporally. However, these characteristics do not exist infinitely for real phenomenon and at certain point, this structure distorts.



### 2.1.2 CHAOS AND RANDOMNESS

The natural laws discovered by Newton and Kepler's represent the ordered behaviour of nature contrary to Chaos where nature does not obey linearity. Chaos does not necessarily represent the high degree of complexity but something does not comply with previously known laws. They are challenging because of their non-linear behaviour as nature does not follow any fixed path to change its states. A chaotic system represents a system with an extreme sensitive dependence on initial conditions [24]. A minute variation in initial conditions may results in diverse outputs. As with an initial measure of 2.0, the output of a chaotic system may be entirely different from that of the same system with an initial measure of 2.00001. Chaos was coined by Edward Lorenz, in 1960 while working on weather forecasting he reached to the conclusion that it is impossible to forecast the weather accurately. Further, his studies led to other issues of what eventually got to be known as the chaos theory [24]. Numerous examples of chaotic systems have been found in ecology, biology, physics, astronomy, computer sciences, fractal images (Mandelbrot set, Sierpinski triangle, and Koch curve), etc. So, nothing in nature seems to be random but only because of the incompleteness set of knowledge. To conclude, a simple process can generate a complete random output; this means that the complexity in nature may be outcome of some simple phenomenon. However, this does not mean that the complex phenomenon is become easy to predict because there are various factors which are complementing the phenomenon hard to control. The characteristic feature of chaotic systems is the fact that randomness creates finally deterministic shapes or trajectories as well invariants

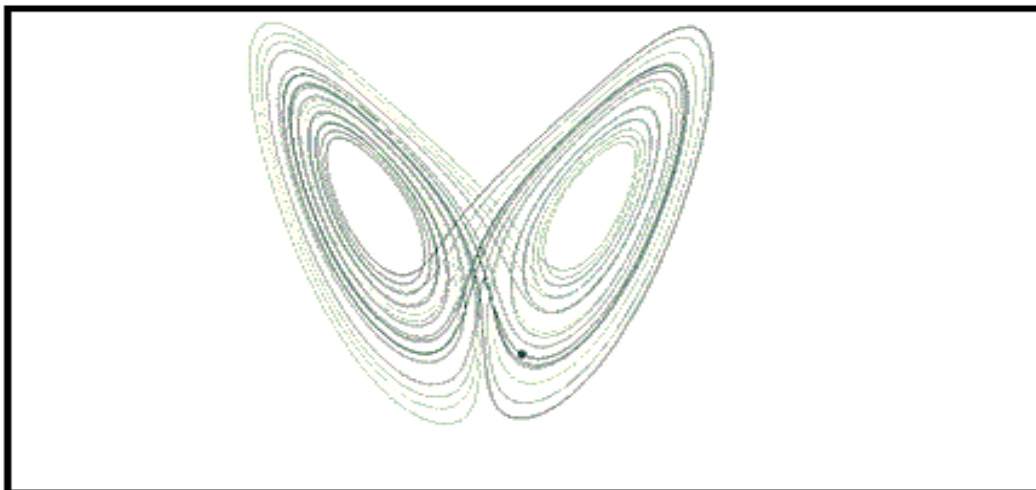


Figure 2.1: Lorenz Attractor When  $p = 28$ ,  $\Sigma = 10$  And  $b = 8/3$  Resembling a Butterfly, Showing How the Chaotic System Forms a Deterministic Shape.

For instance, figure 2.1 is manifesting a form of butterfly, it's made up of the Lorenz attractor .That is a set of chaotic solutions of the Lorenz system, when plotted either forma butterfly or figure eight .Random and chaotic apparently shows that they do not follow any regular pattern but when they zoomed in they follow some common characteristics as Lorenz system does.

### 2.1.3 FRACTALS

Self-similarity can be associated with fractals, which are objects with unchanged appearances over wide range of scales. The concept of fractals has geometrical, statistical as well as dynamic presence. That means, there are fractals of diverse dimensions, e.g., geometrical, statistical and dynamical. Geometrical fractal processes are the Cantor set, Sierpinski triangle, Koch curve. Statistical fractals are the probability density that repeats itself on every scale, like in economics Pareto's law, in linguistics Zipf's law and in sociology Lotka's law [23]. Whereas, dynamical fractal is generated by a low-dimensional dynamical system with chaotic solutions, like in biology Willis law, in medicine cardiac spectrum, mammalian lung and in teletraffic modeling chaotic deterministic maps [24] . Figure 2.2 depicting four levels of iterations of the curve with the same reference shape that is of equilateral triangle. With every iteration the basic reference structure that is triangle in this case is changing its angle from 60 degrees to 120 degrees and makes a complex structure as manifested by left side of figure 2.2.

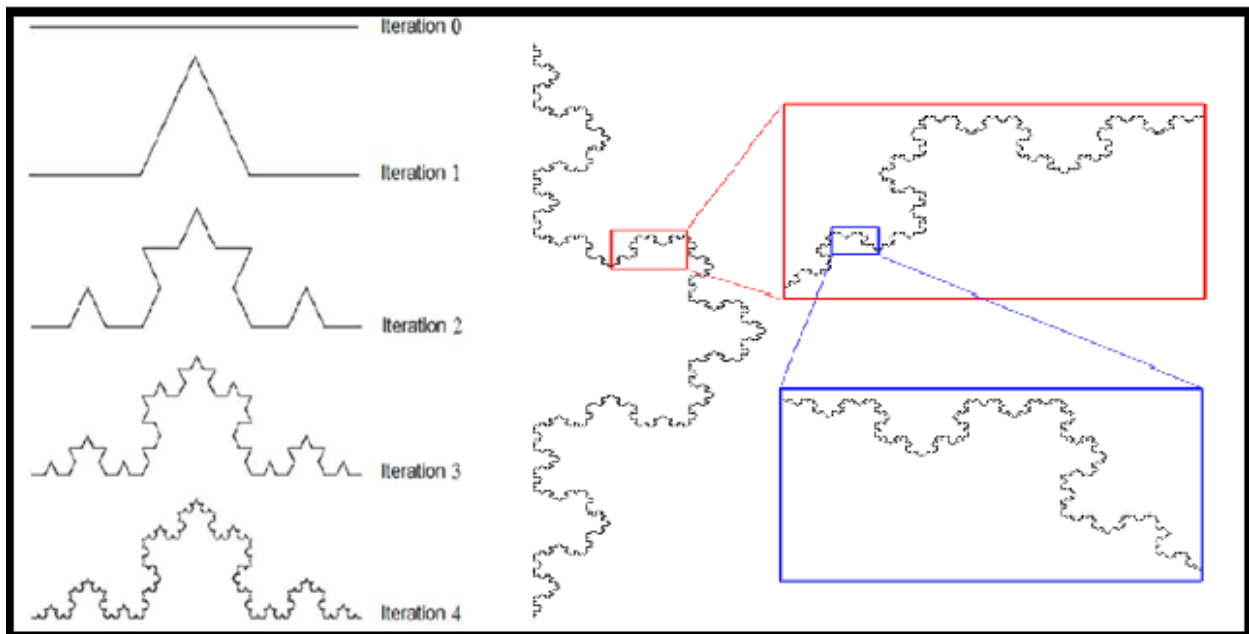


Figure 2.2: Koch Curve with Four Levels of Iteration with Reference Shape of Equilateral Triangle.

Whereas, right side of figure 2.2 is manifesting the self-similarity found when complex structure Koch Curve is breakdown into smaller scales or alternatively after zooming in the basic structure of the curve remains same.

## 2.2 TRAFFIC SELF-SIMILARITY

As Conventional models of traffic have either independent or only short-term auto correlated values. They could be aggregated, by taking simple means of arrival or intervals stream, they tend towards a sequence of random variables, or covariance stationary white noise. A self-similar sequence shows a completely different behaviour. It remains statistically self-similar or asymptotically self-similar. As Figure 2.1 manifest the traffic self-similarity irrespective of the time scale. However, it must be distinguishable from the white noise. The selection below discusses the characteristics of statistical self-similarity in detail.

### 2.2.1 STATIONARY RANDOM PROCESSES

A stationary random process is a stochastic process whose unconditional joint probability distribution remains constant when shifted in time that is time invariance. In other words, mean and variance do not change over time and keep constant under various scales [27]. Mathematically, for a time series  $X = (X_t, t = 0, 1, 2, 3, \dots)$ , if  $X(t) \equiv X(t + \Delta)$  and have same probability distributions then  $X(t)$  is called stationary random process. The arrival of packets in any time  $t$  is a counting process  $N(t)$  that can be represented as:

$$N(t) = \int_0^t dN(x) \quad (2.1)$$

Here  $dN(t)$  depict a point process for packet arrivals. At any point in time  $t$ ,  $X(t)$  is a random variable and its complete form over time is a random process. If  $n$  is the total number of packets for a time series then the processes involve in stationarity are given below. Mean of Stationary random process:

$$\mu = E[X_n] \quad (2.2)$$

Variance of Stationary random process:

$$\sigma^2 = \text{var}[X_n] = E[(X_n - \mu)^2] \quad (2.3)$$

Autocorrelation function for a Stationary random process:

$$\text{Cov}(X_n, X_{n+k}) = E[(X_n - \mu)(X_{n+k} - \mu)] \quad (2.4)$$

Index of Dispersion for packet counts:

$$F(t) = \frac{\text{var}[N(T)]}{N(T)} \quad (2.5)$$

Mean, variance, covariance, autocorrelation and index of dispersions are methods to find the underlying structure of any process statistically. For a stationary process mean, and variance must remain same and autocorrelation exists strongly for any length of time. But such processes lack a characteristic scale as they contain only a particular reference structure.

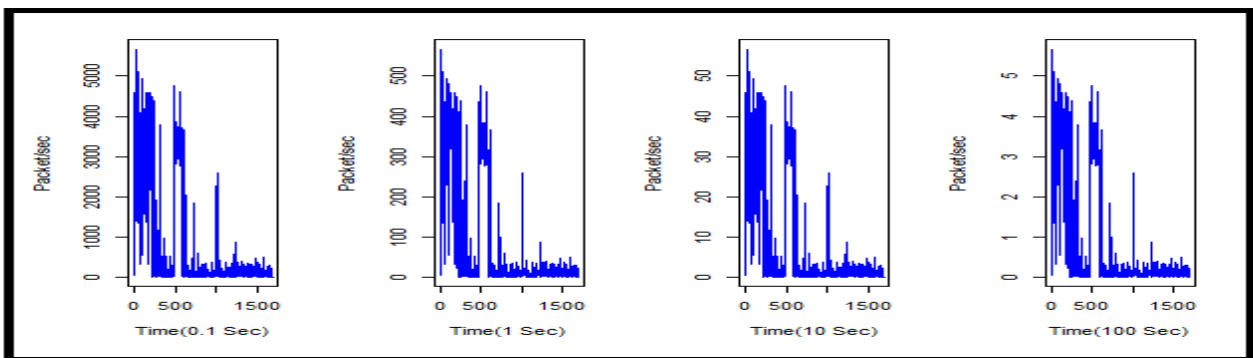


Figure 2.3: Traffic Trace on Four Different Time Scales Depicting Self-Similar Characteristics

Figure 2.3 is manifesting the underlying self-similarity in the traffic obtained after plotting it on multiple time scales such as 0.1 seconds, 1 second, 10 seconds and 100 seconds. This shows the scale invariant property of the data self which remained same at all four scales but number of packets are reduced as time scale become larger, packets ranging from thousands to tens.

### 2.2.2 SELF-SIMILAR PROCESSES

A continuous time stochastic process  $X(t)$  is self-similar if Hurst parameter ( $H$ ) is equal to ( $0.5 \leq H \leq 1$ ). As Leland [7] et al defined the continuous time self-similar process: let  $X$  be a time series  $X = (X_t : t = 0, 1, 2, 3, \dots)$  with autocorrelation function:

$$r(k) \sim k^{-\beta} L(t), \text{ as } k \rightarrow \infty \quad (2.6)$$

Here  $0 < \beta < 1$  and  $L$  is a function slowly varying at infinity and could be defined as  $\lim_{t \rightarrow \infty} \frac{L(tx)}{L(t)} = 1$  for all  $x > 0$ . However, consideration of  $L$  as asymptotically constant keeps the results simple. For the blocks  $m = 1, 2, 3, 4, \dots$  let  $X^{(m)} = X_k^{(m)} : K = 1, 2, 3, \dots$  depict a covariance stationary time series obtained by averaging  $X$  over non-overlapping blocks of size  $m$ . So that

$X_k^{(m)} = \frac{1}{m} (X_{km-m+1} + \dots + X_k)$ ,  $K \geq 1$  is called second order self-similar process with  $H = 1 - \frac{\beta}{2}$  and  $\text{var}(X^{(m)}) = \sigma^2 m^{-\beta}$  so that:

$$r^{(m)}(k) = r(k), k \geq 0 \quad (2.7)$$

$$r^{(m)}(k) = r(k), m \rightarrow \infty \quad (2.8)$$

For any real positive  $a$ ,  $X(t)$  would be  $X(t) = a_H X(at)$ . The mean, variance and autocorrelation of such process would be:

$$E[X(t)] = \frac{E[at]}{a^H} \quad (2.9)$$

$$\text{Var}[X(t)] = \frac{\text{Var}[X(at)]}{a^{2H}} \quad (2.10)$$

$$R_{x(t,s)} = \frac{R_x(at, as)}{a^{2sH}} \quad (2.11)$$

Under such systems as  $H \rightarrow 1$  self-similarity and Long range dependent behavior become more visible. That means that a pattern in the past implies with a large probability, same pattern in future.

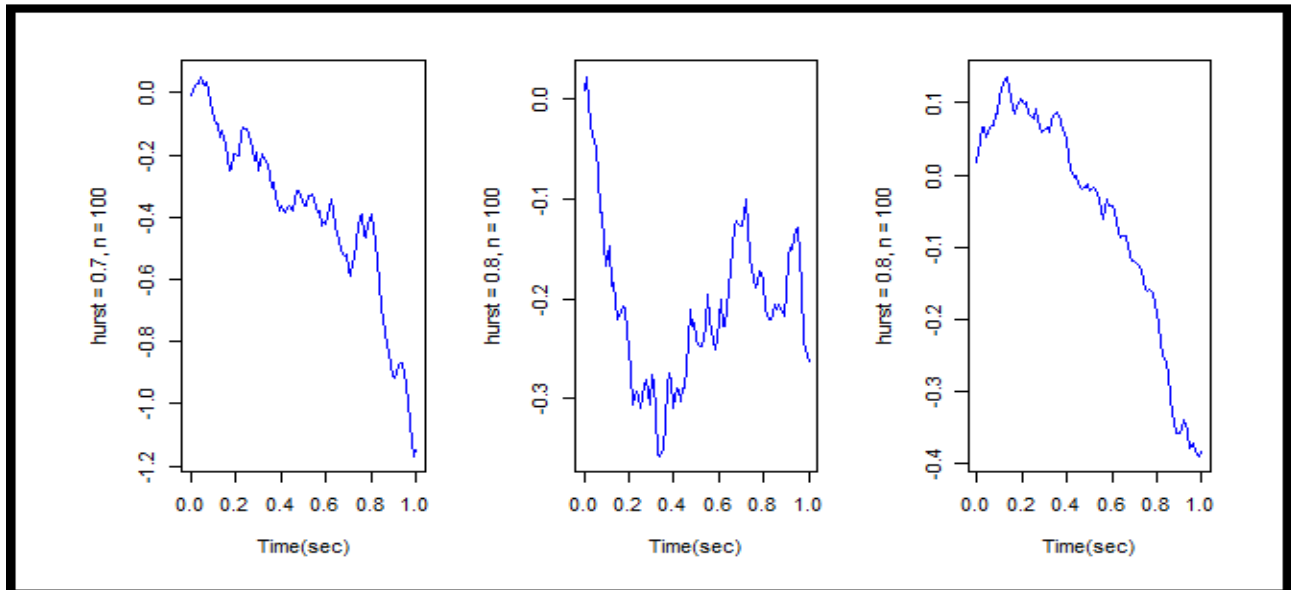


Figure2.4: Fractional Brownian Motion with  $H = 0.7, 0.8, 0.9$  and  $n = 100$

Figure 2.4 manifest a continuous time Gaussian process  $B_H(t)$  with different values of  $H$ , such as 0.7 and 0.8 respectively, but with same number of packets shows a positive correlation over time  $t$ . It is a non-stationary process with stationary increments of order  $n$ .

### 2.3 PROPERTIES OF SELF-SIMILAR PROCESS

There are some properties which self-similar processes manifest which makes them discern among other types of traffic such as Short Range Dependence (SRD). These properties are described in the following section.

#### 2.3.1 SLOWLY DECAYING VARIANCE

Generally variance is defined as the measure of how far a series could spread from their mean value. Slowly decaying variance means that it would follow power law decay with decaying parameter of  $\beta$  [28] contrary to sample size. So that variance for time series over  $m$  non-overlapping blocks could be written as:

$$\text{var}(X^{(m)}) = \sigma^2 m^{-\beta} \quad (2.12)$$

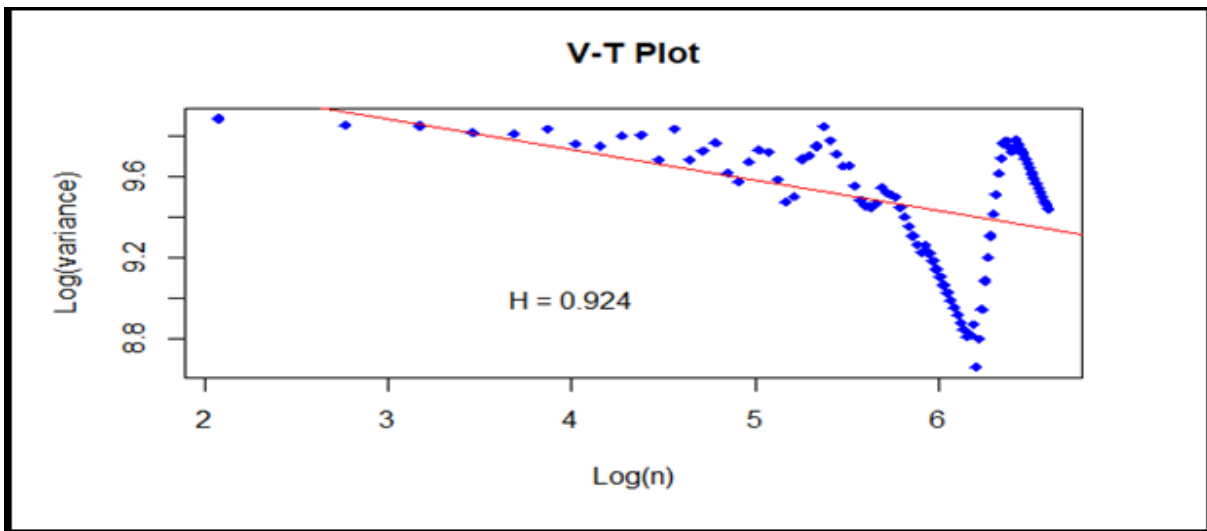


Figure 2.5: Variance Time Plot Showing the Decaying Constant  $\beta = -0.152$  Having Negative Slope and  $H = 0.924$

The slowly decaying variance property can be easily detected by index of dispersion method or by variance time log-log plot [29]. Figure 2.5 manifest the log- log plot of a time series with negative slope of decaying constant manifesting the self-similarity exists.

### 2.3.2 HURST EFFECT

While studying river flows of Nile Hurst discovered this effect [33]. Hurst studied the rescaled Adjusted range R/S statistics. If  $X_i$  represents the river flow into a reservoir and differences between the  $X_i$  and the mean flow is  $X(d)$ , represents the constant outflow downstream. The cumulative sequence  $W_k$  found from these differences gives the total outflow over the year  $k$ . The range of  $W_k$  is  $R(d)$  and can be calculated by:

$$R(d) = \max(0, W_1, W_2, W_3, \dots, W_d) - \min(0, W_1, W_2, W_3, \dots, W_d) \quad (2.13)$$

$$W_k = \sum_{j=1}^k (X_j - K\bar{X}(d)) \quad (1 \leq k \leq d) \quad (2.14)$$

The relation between Standard deviation and  $R(d)$  would be:

$$\frac{R(d)}{S(d)} \sim C_1 d^{\frac{1}{2}} \quad , d \rightarrow \infty \quad (2.15)$$

Or it can be put it in another way:

$$\frac{R(d)}{S(d)} \sim C_2 d^H \quad , d \rightarrow \infty, \frac{1}{2} < H \leq 1 \quad (2.16)$$

For various naturally occurring phenomenon  $H$  appears to around 0.75. As  $H$  is closer to 1 the presence of self-similar pattern becomes certain.

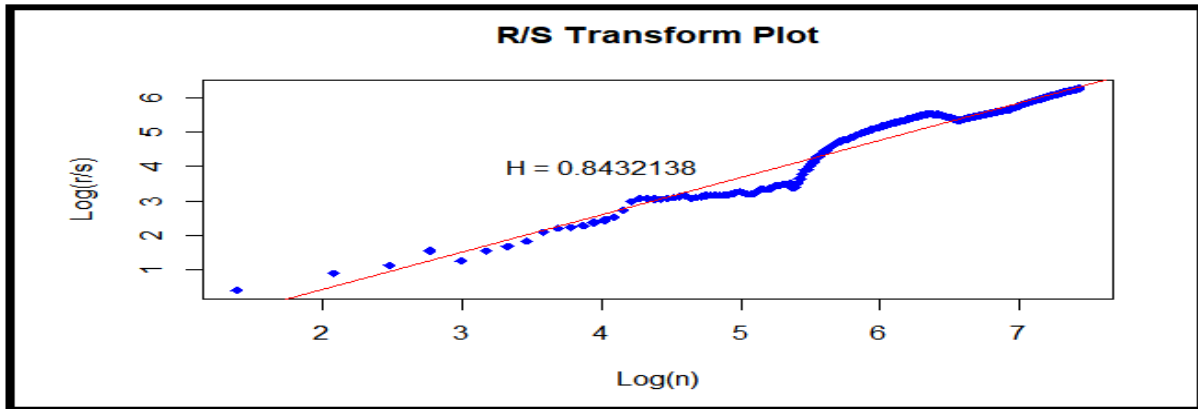


Figure 2.6: R/S statistic Plot Showing  $H = 0.84$  for a Traffic Trace.

Figure 2.6 shows R/S over a trace of internet traffic. The value of  $H$  founded is 0.84 which manifest that data is highly self-similar in the captured data trace. R/S transform is one of the best methods for finding Hurst effect when value is in between 0.5 to 1, the data is self to self-similar.

### 2.3.3 1/F NOISE

1/f Noise depicts property of LRD in frequency domain. The Power Spectral Density (PSD) of LRD processes diverges at low frequencies contrary to SRD processes where the spectral density remains flat at low frequencies. In short, the PSD of LRD possesses a power-law behavior near the frequency origin.

$$S(w) \sim \frac{1}{|w|^\gamma} \quad (2.17)$$

When  $|W| \rightarrow 0$  and  $0 < \gamma < 1$ ,  $\gamma$  and  $\beta$  could be related by:

$$\gamma = 1 - \beta = 2H - 1 \quad (2.18)$$

SRD processes have finite power spectral densities even when  $|W| \rightarrow 0$  or  $\gamma \rightarrow 0$  but LRD behaves in a hyperbolic decay function and has large summable autocorrelation function.



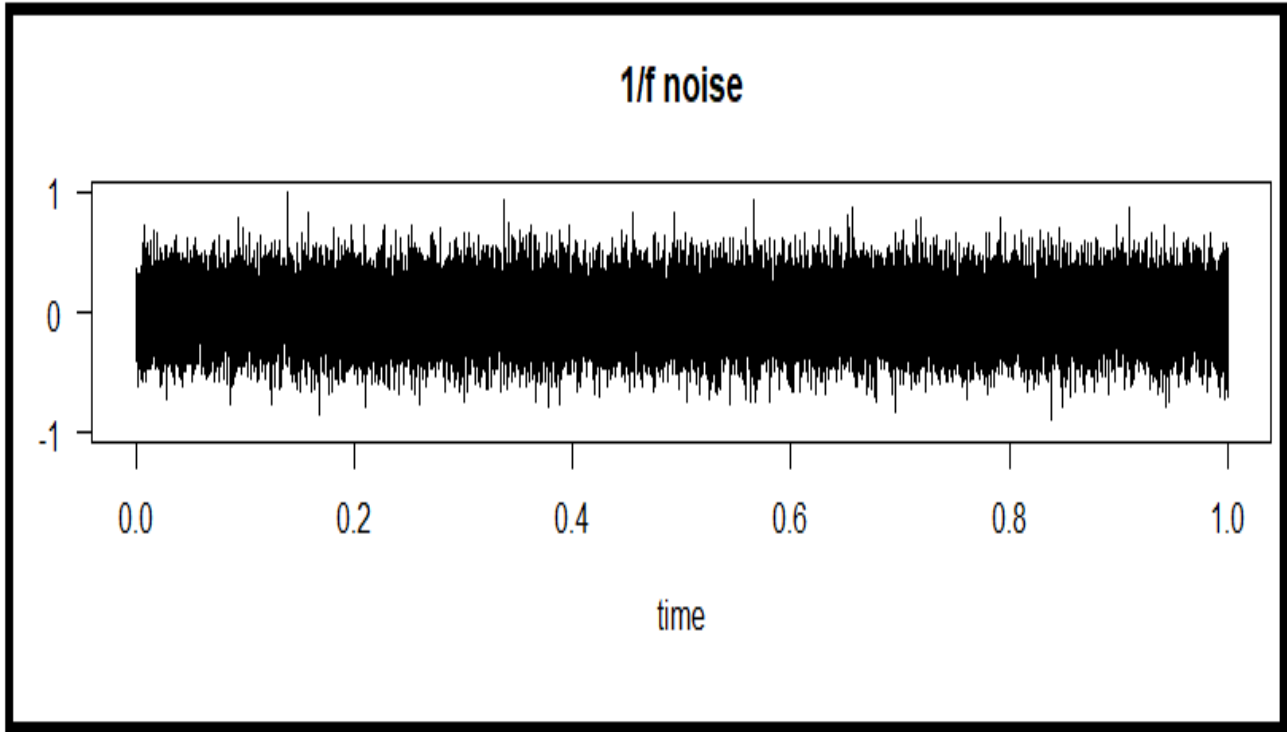


Figure 2.7: 1/f Noise of Traffic Trace Showing the Basic Characteristic of Apparently Random Fluctuations but Follow a Self-Similar Pattern

Figure 2.7 shows the 1/f noise of a traffic trace. It shows apparently random fluctuations but that are stochastically self-similar and decay is power law decay which follows the hyperbolic trajectory if plotted on frequency scale.

### 2.3.4 HEAVY- TAILED DISTRIBUTIONS

Heavy-tailed distributions property depends upon the marginal amplitude distributions of  $X_n$  contrary to previously discussed scaling factors such as autocorrelation etc. One of such example is and ON/OFF periods in packet switching networks [33]. Therefore,  $X$  is said to be heavy-tailed if its complementary cumulative distribution function (CCDF) decay follows power-law distribution as represented by equation (2.19).

$$P\{X \geq x\} \sim x^{-\alpha} \quad (2.19)$$

Here,  $\alpha$  is called tailed index, as  $x \rightarrow \infty$  and  $a > 0$ . For  $0 < \alpha < 1$  all moments of  $X$  are infinite and  $n$ -th moment is infinite for  $n > \alpha$ . Pareto distribution is an epitome of heavy tailed distribution in which tail end follows a power law decay. Not necessarily heavy tailed distributions are self-similar but self-similar processes have very much in common as that of heavy tailed distribution.

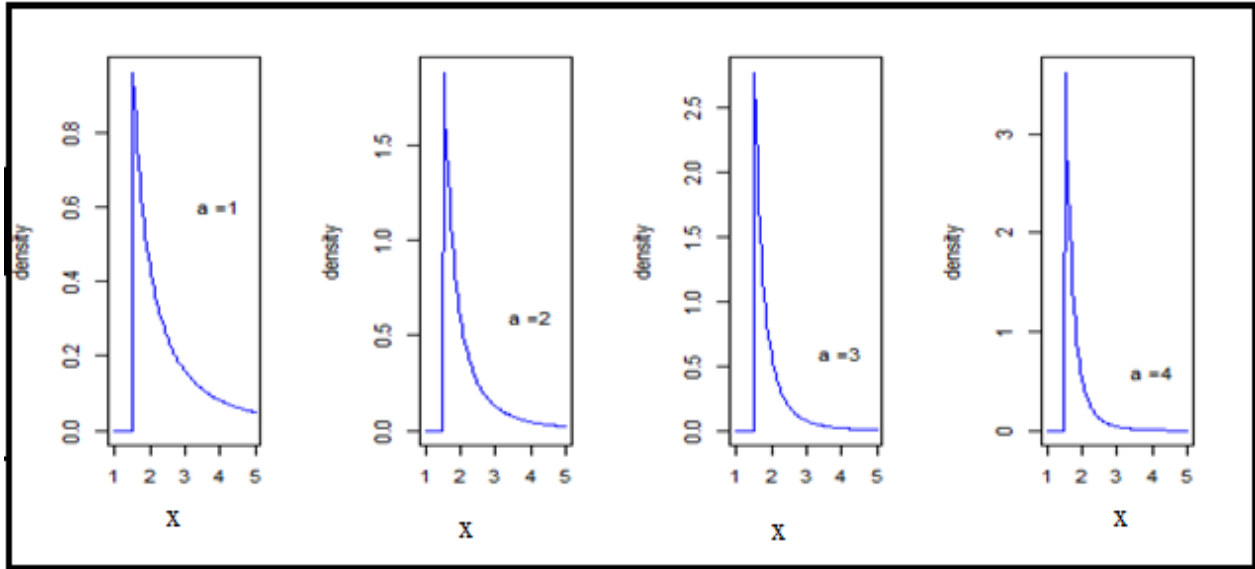


Figure 2.8: Probability Density Function of Pareto Distribution

Figure 2.8 is one of the manifestations of heavy tailed distributions that are Pareto distribution whose decay follows power law for different values of  $a$  is shape parameter handles the shape of distribution instead of scaling behaviour. Discusses different types of heavy tailed distribution in which power law distributions has given significant importance.

## 2.4 SELF-SIMILAR TRAFFIC MODELS

There are many models that can generate self-similar data. Following are few models that can generate self-similar traffic when used with varying degree of complexity and mathematical models.

### 2.4.1 AGGREGATE TRAFFIC MODELS

#### 2.4.1.1 FRACTIONAL BROWNIAN MOTIONS

Brownian motion  $B(t, w)$  is a random function of independent Gaussian increments. The motion  $B_H(t_2, w) - B_H(t_1, w)$  has zero mean and variance. Fractional Brownian motion (fBm) with  $H$  increments has a Gaussian marginal distribution with zero mean and variance of  $|t_2 - t_1|^{2H}$ . In this case the Holmgren-Riemann-Lowville fractional integral [35] would be :

$$B_H(t, w) = \frac{1}{\Gamma(H + \frac{1}{2})} \int_0^t (t-s)^{H-\frac{1}{2}} dB(s, w) \quad (2.20)$$

$$B_H(t_2, w) - B_H(t_1, w) = \frac{1}{r(H+\frac{1}{2})} \int_{-\infty}^{t_2} (t_2 - s)^{H-\frac{1}{2}} dB(s, w) - \int_{-\infty}^{t_1} (t_1 - s)^{H-\frac{1}{2}} dB(s, w) \quad (2.21)$$

Here,  $0 < H < 1$  and  $r(H + \frac{1}{2})$  is function when  $H < \frac{1}{2}$  the motion becomes ordinary repeated integrals. Fractional Gaussian Noise is a generalized way of differentiation of Fractional Brownian motion. It has the autocorrelation function  $r(k)$ . Autocorrelation function equation 2.22 help to analyze how a series can relate to its past at different lags.

$$r(k) = 1/2(|k + 1|^{2H} - 2|k|^{2H} + |k - 1|^{2H}) \quad (2.22)$$

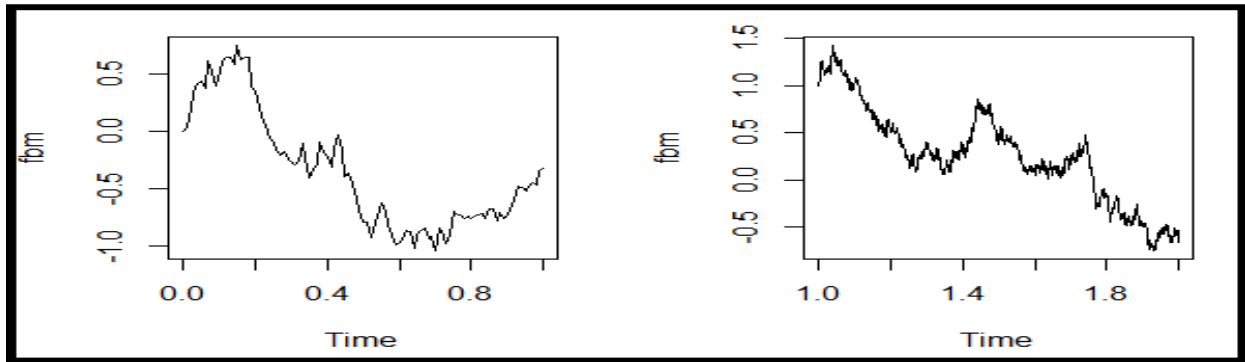


Figure 2.9: Fractional Brownian Motion

Figure 2.9 shows the fractional Brownian motions of a time series X. It can put into another way that an fBm is a moving average of  $B(t, w)$  where past increments are weighted by  $(t - s)^{H-1/2}$ . The two graphs at different time scales manifest the incremental behaviour of Brownian motion.

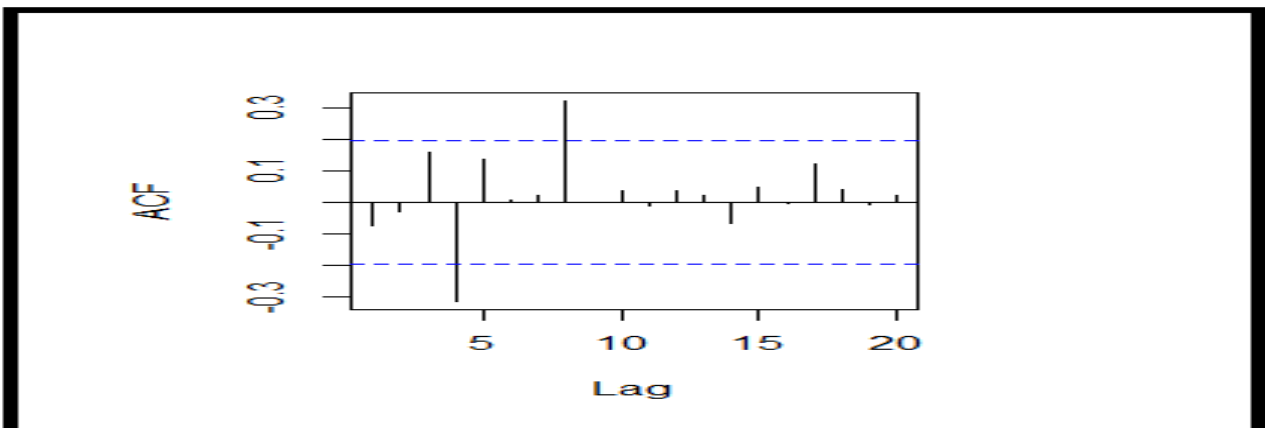


Figure 2.10: Autocorrelation Function of Time Series X at various lags

$$F(x) = 1 - \left( \frac{a(x-c)}{b} \right)^{\frac{1}{a}}, a \neq 0 \quad (2.28)$$

Figure 2.10 manifest the autocorrelation function of a self-similar time series at various lags ranging from 5 to 20 their ACF values are in range of -0.3 to 0.3. Shows that how the self-similarity persist in the data set at such a wide variety of lags.

#### 2.4.1.2 FRACTIONAL ARIMA MODELS

Auto Regressive Integrated Moving Average or Fractional ARIMA is a generalization of the Box Jenkins ARIMA{p, d, q} model [35 ] d is non-integer value, for  $0 < d < 1/2$  the process would be asymptotically second-order self-similar with  $H = d + 1/2$ . It supports long memory process with power law decay. An ARIMA {p, d, q} process is given by:

$$\phi(B)\nabla^d x_k = \Theta(B)\varepsilon_k \quad (2.23)$$

Here  $\phi(B)$  and  $\Theta(B)$  are polynomials and fractional differential operator is defined as  $\nabla^d$  :

$$\Delta^d = (1 - B)^d = 1 - dB - \frac{1}{2}d(1-d)B^2 - \frac{1}{3}d(1-d)(2-d)B^3 - \dots \quad (2.24)$$

It advantage over fractional Gaussian noise is that it can also regard short term autocorrelation long with simple spectral density.

#### 2.4.2 SINGLE SOURCE MODEL

##### 2.4.2.1 PARETO DISTRIBUTION

Pareto distribution is a power curve with two parameters, namely the shape parameter  $\alpha$  and the lower cutoff parameter  $\beta$ . The Pareto cumulative distribution function (cdf) and probability distribution function (pdf) can be given by:

$$F(x) = 1 - \left( \frac{\beta}{x} \right)^\alpha \quad (2.25)$$

$$f(x) = \frac{\alpha}{\beta} \left( \frac{\beta}{x} \right)^{\alpha+1} \quad (2.26)$$

A parameter  $\alpha$  is related to H as:

$$H = (3 - \alpha) / 2 \quad (2.27)$$

##### 2.4.1.2 GENERALIZED PARETO DISTRIBUTION

A 3-parameter Generalized Pareto (GP) distribution can be defined as:

$$F(x) = 1 - \exp\left(-\frac{x-c}{b}\right), a = 0 \quad (2.28)$$

Here  $c$  is a location parameter,  $b$  is a scale parameter,  $a$  is shape parameter. By using this Pareto distributions could be obtained for  $a < 0$ . Further the 2- parameter GP distribution can be obtained when  $c=0$ , the exponential distribution for  $a=0$  and  $c=0$ , and the uniform distribution on  $[0, b]$  for  $c=0$  and  $a=1$ .

## **2.5 SUMMARY OF THE CHAPTER**

Self-similar process gained importance as Leland, Crovella, Willinger; Taqqu et al [7] published their seminal works on the nature of self-similarity, its presence in the internet traffic and its dominance over tcp flows. This chapter has tried to give a brief insight of the self-similar processes. As the topic is really vast and it is next to impossible to define every aspect of it in this project. However, a mathematical touch of self-similarity, fractal and, chaos has given in this chapter. Random processes and particular properties self-similarity such as slowly decaying variance, Hurst effect and i/f noise are also described in this chapter. The most significant part of this chapter is the definition of models of self-similar traffic such as aggregated traffic models, Fractal Brownian Motions, Fractal ARIMA models, Pareto distributions and Generalized Pareto distributions.

## CHAPTER 3

### HURST PARAMETERS ESTIMATION TECHNIQUES

#### 3.1 INTRODUCTION

Parsimonious network models require minimum number of parameters to control their behaviour. Hurst parameter (H) characterizes the presence of LRD. H is closely associated with power law, long memory, fractal, fractional calculus and chaos theory [24, 25]. In order to estimate H there are number of estimation techniques exists but their accuracy and validity may differ. As their efficiency is disturbed by the presence of white noise, pink noise, and others. The robustness of any estimation techniques requires a thorough analysis. In this chapter one of such analysis is provided by finding the validity and accuracy of twelve estimators for H. The best-known Hurst estimator is the rescaled range (R/S) method that was first proposed by Hurst [26]. The twelve methods are R/S method, aggregated variance method, absolute value method, variance of residuals method, Periodogram method, modified Periodogram method, local Whittle method, diffusion entropy method; Kettani and Gubner's method, Abry and Vetch's method, Koutsoyiannis' method, and Higuchi's method. However, four of them are implemented for this project.

#### 3.2 METHODS OF ANALYZING SELF-SIMILARITY

There are various methods for calculating and analyzing self-similarity in a traffic trace. Most famous are twelve of them as discussed previously. They work on time and frequency domains respectively. Following are the four methods widely used to measure self-similarity.

##### 3.2.1 R/S TRANSFORM

R/S method is one of the most well-known estimators. For a time series X of length N, R(n) is the range of the data aggregated over m non-overlapping blocks (m) and S(n) is standard deviation. For fGN or FARIMA processes:

$$E[R/S(n)] \sim Cn^H, \quad n \rightarrow \infty \quad (3.1)$$

$$\text{Range} = \text{Max}(z_1, z_2, z_3, \dots, z_t) - \text{Min}(z_1, z_2, z_3, \dots, z_t) \quad t = 1, 2 \dots n \quad (3.2)$$

Here n is the number of observations or packets in the trace file, and C is a positive integer constant, whereas, Z is the cumulative derivate and could be given by:

$$z_t = \sum_{i=1}^t Y_t, t = 1, 2, 3, \dots, n \quad (3.3)$$

$Y_t$  is adjusted mean of  $X_i$  time series.  $H$  can be estimated by a slope of log-log plot of  $R/S$  over  $n$ .

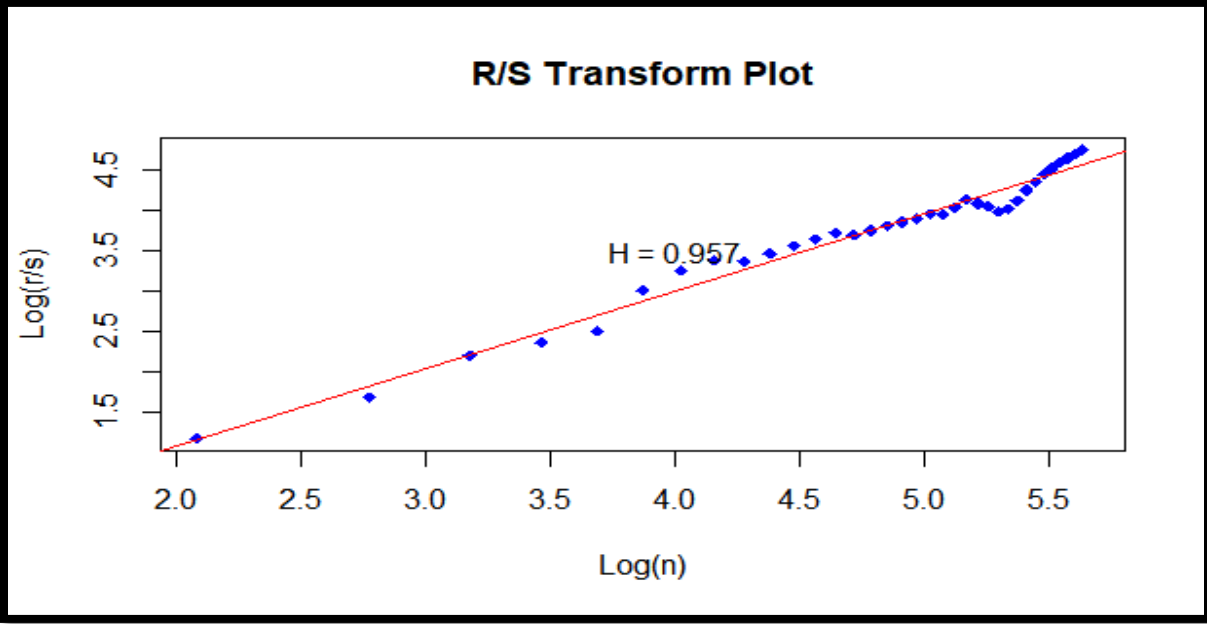


Figure 3.1: R/S Plot of a Traffic Trace along with a Slope Depicting  $H= 0.957$

Figure 3.1 manifesting the log-log plot of a traffic trace of  $R/S$  transform to measure the self-similarity of the trace. On x-axis the number of packets in a trace file are plotted after taking their log whereas on y-axis the log of fraction of range over standard deviation is plotted after taking its log over non-overlapping block of size  $m$ .  $m$  could be ( $m = 8, 16, 32, 64, \dots$ ) etc. The blue dots in the diagram are showing time series whereas the red is regression fit for series. Its slope gives the value of  $H$  for this method directly.

### 3.2.2 WAVELET BASED H ESTIMATOR

The wavelet-based  $H$  estimator is based on a spectral density and frequency obtained by performing a time average of the wavelet detail coefficients  $d(i, j)$  at a given scale:

$$S_x = \frac{1}{n_j} \sum_j |d(i, j)|^2 \quad (3.4)$$

Here  $n_j$  is the number of wavelet coefficients at  $i$  scale and  $j$  is the octave of wavelet, and detail coefficient could be defined as:

$$d(j, k) = \frac{X_{2k}^{j-1} - X_{2k+1}^{j-1}}{2^{j/2}} \quad (3.5)$$

The energy function of wavelet transform would be:

$$\text{Energy} = \frac{\sum_k d(j, k)^2}{N_j} \quad (3.6)$$

Here,  $k=1 \dots N$ , the plot of Energy vs. octave  $j$  is plotted to estimate  $H$  after taking slope of the distribution could be computed by slope parameter  $\beta$ .

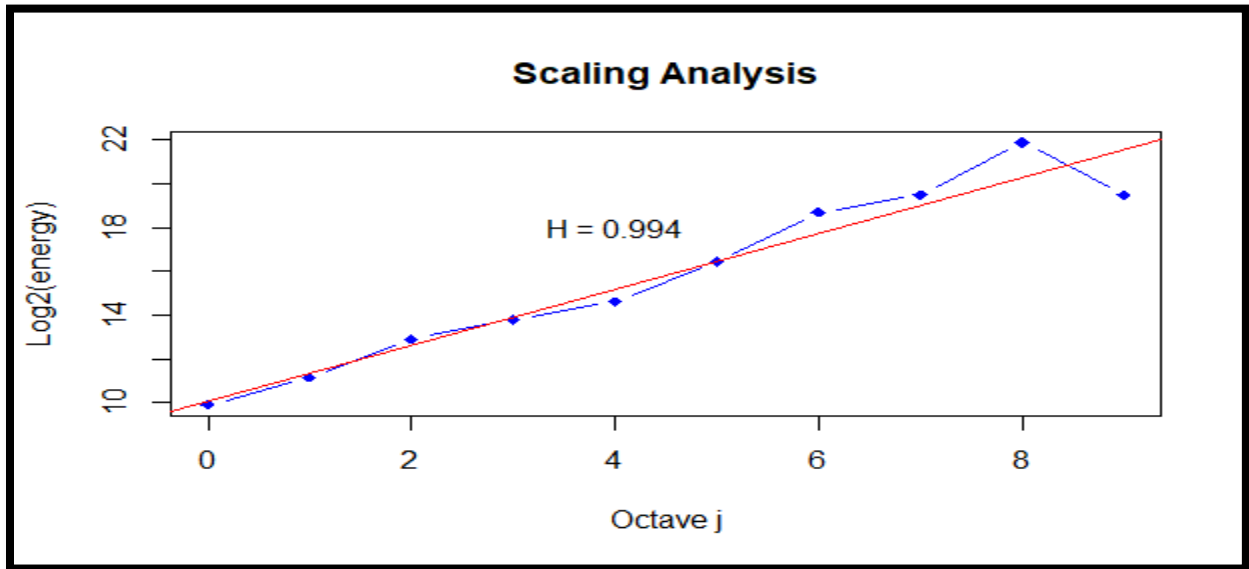


Figure 3.2: Wavelet Estimator: Energy vs. Octave  $j$  Plot to Estimate  $H$  by Slope  $\beta$  of Regression Line;  $H= 0.994$

Figure 3.2 depicting a plot of Energy vs. octave  $j$  to estimate  $H$  by the slope of the regression line. The blue dots are manifesting the energy filled in every octave by taking the difference between two successive detail points in the time series of count data of internet traffic trace. When these blue lines are plotted against the octave  $j$  they manifest the inherent scaling factor in the wavelet transform method.



### 3.2.3 PERIODOGRAM BASED ESTIMATORS

The Periodogram method use spectral density and frequency.

$$I(\varepsilon) = \frac{1}{2\pi N} \left| \sum_{j=1}^N X_j e^{ij\varepsilon} \right|^2 \quad (3.7)$$

Here  $j$  depicts the frequency  $X_j$  is the input series,  $I(\varepsilon)$  is an estimate of the spectral density of the series. Periodogram of a LRD process should be proportional to  $|\lambda|^{1-2H}$  close to the origin, so the log-log plot of  $I(\varepsilon)$  should have a slope of  $1-2H$ .

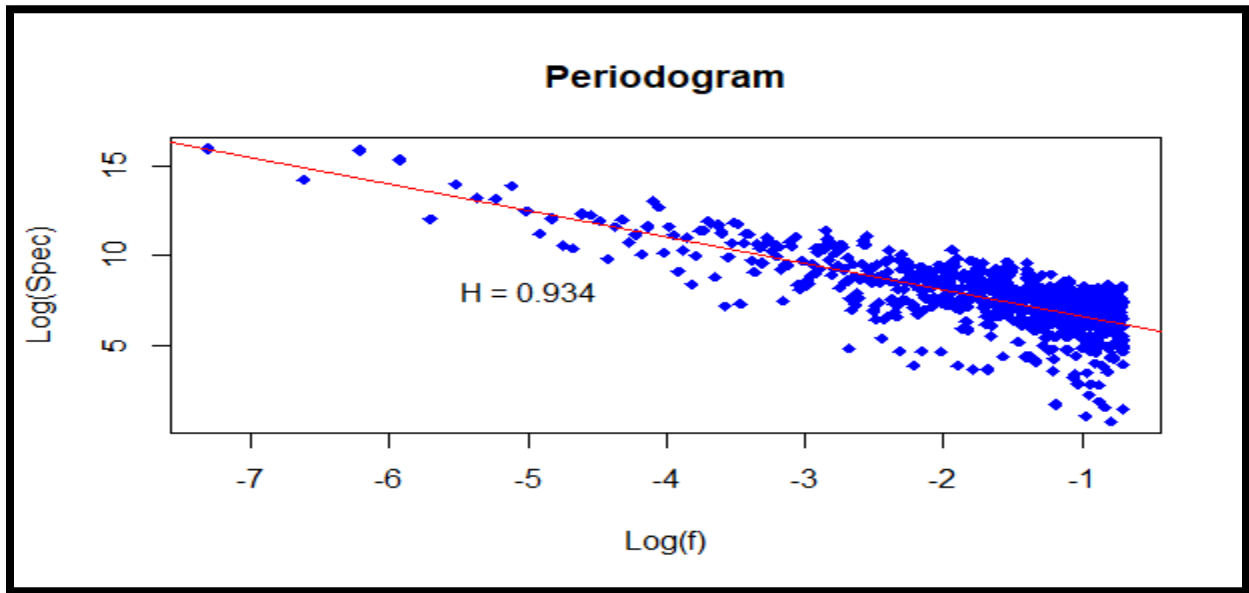


Figure 3.3: Periodogram: Power Spectral Density vs. Frequency with Slope,  $H= 0.934$

A periodogram manifest the spectral density of a time series signal. Figure 3.3 manifest the relation between spectral density and frequency of a signal when plotted against each other and fitted a regression model gives the value of Hurst parameter  $H$  which is 0.934.

### 3.2.4 VARIANCE TIME METHOD

Variance time method works on aggregated series of data if a series is denoted by  $X$  of length  $N$  then the aggregated time series would be:

$$X^{(m)}(K) = \frac{1}{m} \sum_{i=(K-1)m+1}^{km} X(i), K = 1, 2, \dots \quad (3.8)$$

The variance for every successive block would be:

$$X^{(m)}(K) = \frac{1}{m} \sum_{i=(K-1)m+1}^{Km} X(i), K = 1, 2, \dots \quad (3.9)$$

As long range dependence make the time series follow the power law. Variance should asymptotically proportional to  $m^\beta$ , the variance in such case would be :

$$\text{var}(X^{(m)}) = \sigma^2 m^{-\beta}, m \rightarrow \infty, -1 \leq \beta < 0 \quad (3.10)$$

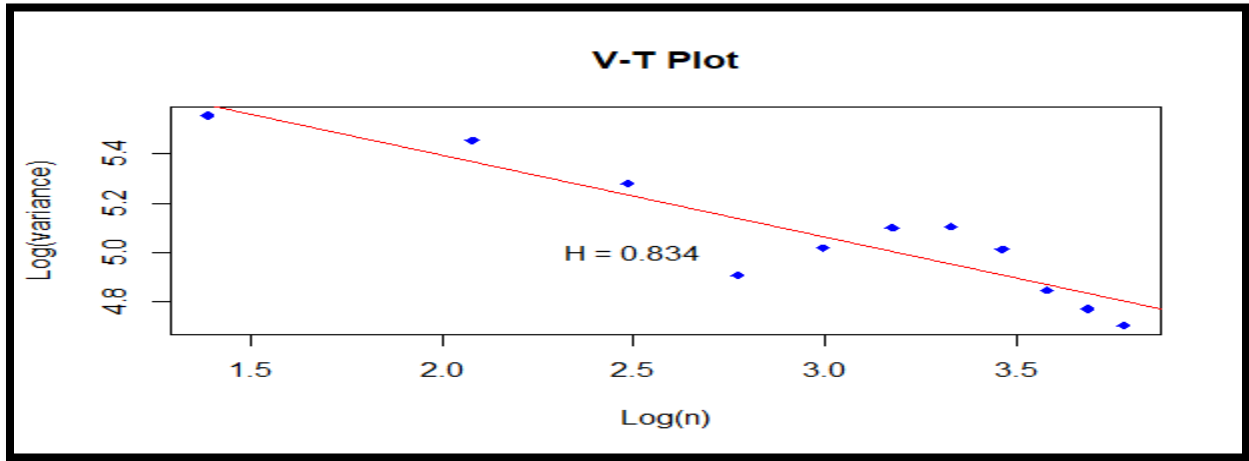


Figure 3.3: Variance Time log-log Plot for a Traffic Trace; the Regression Line Giving the Value of  $H = 0.834$

Figure 3.3 showing the variance time plot of a traffic trace with a regression line and value of  $H = 0.834$ . Variance measure the spreadness of data around the mean value but when traffic is self-similar variance goes to infinity as size of data increases.

### 3.3 MEAN SQUARE ERROR AND R/S TRANSFORM:

Mielniczuk has used the method of mean square error to correct the estimated value of  $H$  by R/S algorithms. Means Square Error can be given by:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{Y})^2 \quad (3.11)$$

Here  $Y$  is the time series and  $n$  is the number of observations in it. The correctness of MSE is another debate however the results provided by it is considered less accurate as compare to regression model. Figure 3.4 is showing the plots of  $H'$  estimated by R/s algorithm and  $h$  after MSE correctness , the meager difference between two plots make this method worth considering for calculation of hurst parameter. Table 3.1 shows the value of  $H$  calculated by R/S transform for the trace files mentioned in chapter 1 in

the section 1.5 descriptions of the traffic trace, the files are divided in such a way that twenty series of count data could be extracted and then their respective value of Hurst parameter is calculated, its MSE and then the corrected value of H. An average difference of 0.19 percent is observed. This means that this transform can perform better to estimate H' as compare to other methods.

**Table 3.1. Estimated and Corrected Value of H by R/S Transform after MSE Analysis**

No.	R/S Estimated H'	Mean Square Error	Corrected H
1	0.781	0.07	0.702
2	0.800	0.08	0.719
3	0.692	0.06	0.622
4	0.843	0.08	0.757
5	0.841	0.08	0.622
6	0.842	0.08	0.757
7	0.843	0.08	0.756
8	0.692	0.06	0.757
9	0.634	0.06	0.757
10	0.758	0.07	0.622
11	0.959	0.09	0.752
12	0.769	0.07	0.573
13	0.843	0.08	0.782
14	0.856	0.08	0.791
15	0.524	0.05	0.455
16	0.63	0.06	0.578
17	0.736	0.07	0.695
18	0.834	0.08	0.791
19	0.86	0.08	0.752
20	0.692	0.06	0.551

Table 3.1 manifests the values of H before and after the correction method that is MSE is applied. While doing so, a difference of 0.19 per cent is observed, between the values of H for a same time series of data. Showing how robust could be the R/S method for estimating the value of H.

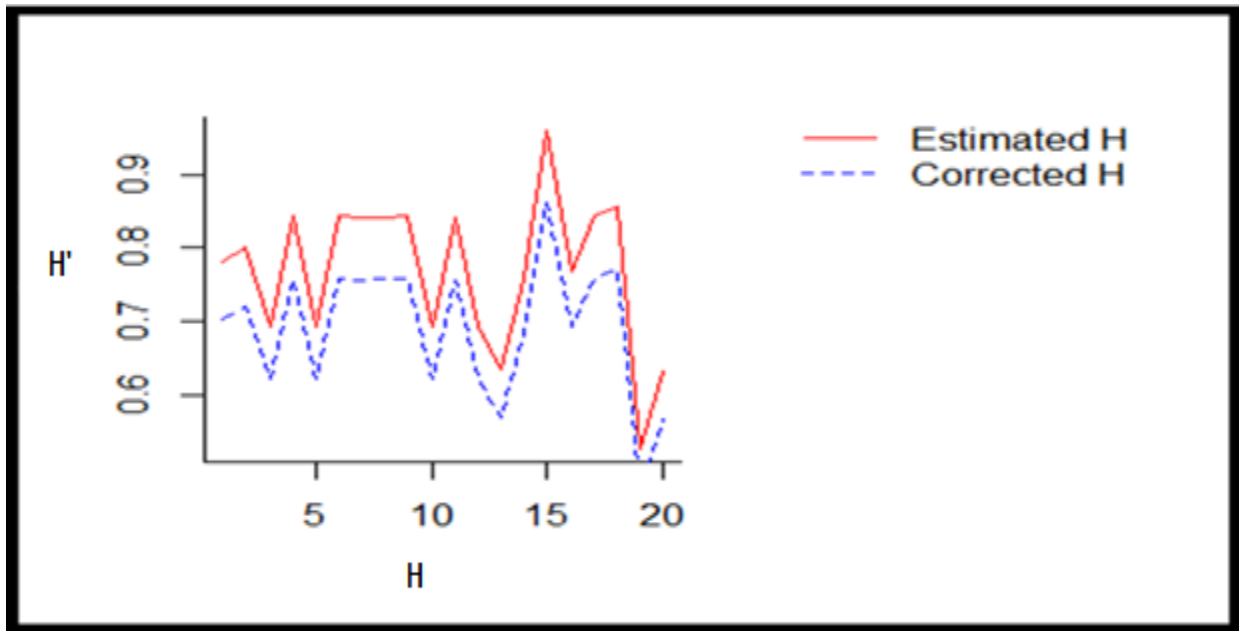


Figure 3.4: Estimated Value of H' by R/S Transform and Corrected Value of H after Error Calculation by MSE of 0.19 Percent

The value of Hurst parameter in Figure 3.4 is of estimated value of H' by R/S Transform and Corrected Value of H after Error Calculation by MSE of 0.19 Per cent. The blue lines shows that the value of H after correcting while the red line show the value of H when no error calculation is done. For R/S transform the difference between the two values is around 0.19 per cent which is a quite minute percentage as compare to periodogram or variance-time method.

### 3.4 MEAN SQUARE ERROR AND WAVELET TRANSFORM METHOD:

The error percentage calculated by this method shows a very little deviation from original value A wavelet transform  $S_x = \frac{1}{n_j} \sum_j |d(i,j)|^2$  and  $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$  are two methods employed in this analysis for the estimated and corrected value of H . As it is defined in the earlier sections that wavelets estimation is the technique for estimating H with inherent scaling capabilities . Table 3.2 shows the value of H calculated by Wavelet transform for the trace files mentioned in chapter 1 in the section 1.5 descriptions of the traffic trace, the files are divided in such a way that twenty series of count data could be extracted and then their respective value of Hurst parameter is calculated, The difference between the two values of H must be very small.

**Table 3.2. Estimated and Corrected Value of H by Wavelet Transform after MSE Analysis**

No.	Wavelet Estimated H'	Mean Square Error	Corrected H
1	0.702	0.024	0.701
2	0.719	0.024	0.710
3	0.622	0.023	0.620
4	0.757	0.012	0.746
5	0.622	0.021	0.601
6	0.757	0.014	0.744
7	0.756	0.024	0.725
8	0.757	0.024	0.725
9	0.622	0.013	0.639
10	0.755	0.014	0.710
11	0.622	0.021	0.624
12	0.570	0.01	0.569
13	0.681	0.16	0.660
14	0.862	0.19	0.870
15	0.757	0.016	0.739
16	0.769	0.024	0.750
17	0.573	0.019	0.560
18	0.567	0.016	0.760
19	0.869	0.018	0.850
20	0.692	0.019	0551

Table 3.2 manifests the values of H before the correction method is applied and after it is applied .while doing so a difference of 0.025 per cent is observes between the values of H for a same time series of data. Showing how resilient could be the Wavelet transform method for estimating the value of H.

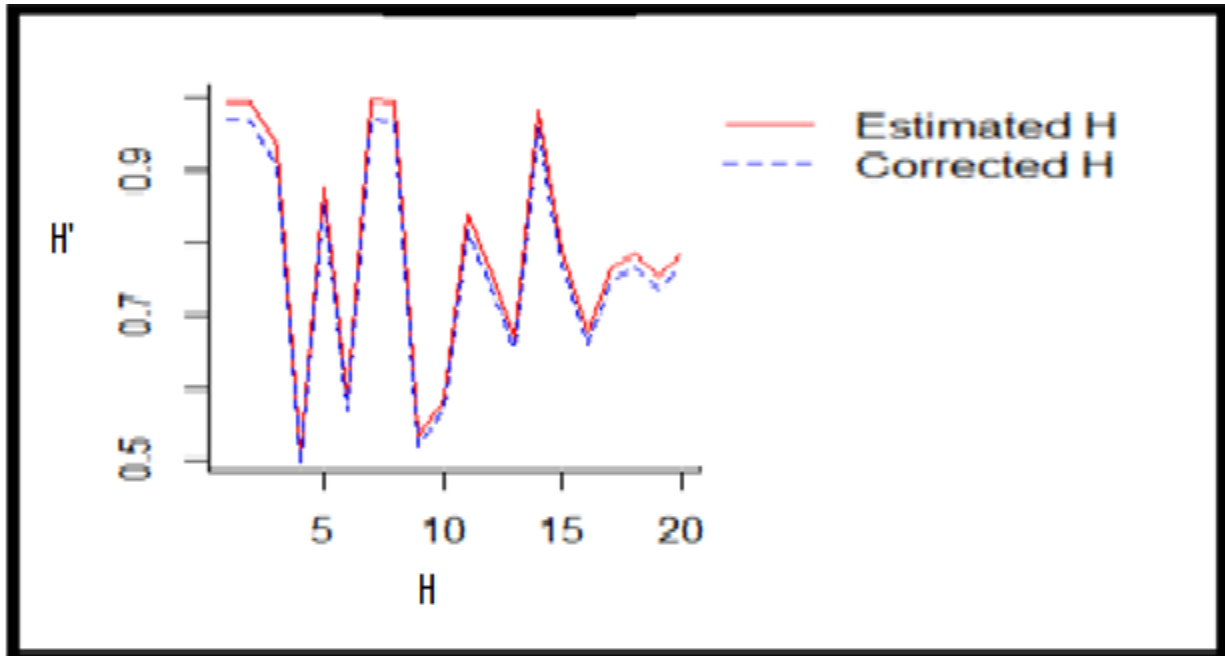


Figure 3.5: Estimated Value of H' by Wavelet Transform and Corrected Value of H after Error Calculation by MSE of 0.025 Percent

The value of H in Figure 3.5 is plotted twice one is before passing through MSE calculations and second time after it. The blue lines shows that the value of H after correcting while the red line show the value of H when no error calculation is done. For wavelet transform the difference between the two values is around 0.025 per cent which is a quite minute percentage as compare to R/S transform, periodogram or variance-time method.

### 3.5 MEAN SQUARE ERROR AND PERIODOGRAM

The estimation of Hurst parameter by Periodogram  $I(\epsilon) = \frac{1}{2\pi N} |\sum_{j=1}^N X_j e^{j\epsilon}|^2$  and its error correction by MSE  $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$  manifested by table 3.3 and figure 3.6 shows the corrected and biased Hurst parameter values with an error percentage of 0.20 percent make the performance of this method little biased for the calculation of Hurst parameter. Table 3.3 shows the value of H calculated by Periodogram for the trace files mentioned in chapter 1 in the section 1.5 descriptions of the traffic trace, the files are divided in such a way that twenty series of count data could be extracted and then their respective value of Hurst parameter is calculated.

**Table 3.3. Estimated and Corrected Value of H by Periodogram after MSE Analysis**

No.	Periodogram Estimated H'	Mean Square Error	Corrected H
1	0.654	0.014	0.639
2	0.673	0.015	0.657
3	0.949	0.021	0.927
4	0.946	0.02	0.924
5	0.897	0.02	0.876
6	0.923	0.020	0.902
7	0.989	0.0208	0.813
8	0.899	0.022	0.766
9	0.991	0.020	0.878
10	0.671	0.0221	0.568
11	0.966	0.022	0.856
12	0.967	0.021	0.873
13	0.890	0.198	0.845
14	0.645	0.0144	0.670
15	0.787	0.017	0.630
16	0.989	0.012	0.869
17	0.675	0.019	0.601
18	0.567	0.022	0.560
19	0.866	0.019	0.850
20	0.987	0.015	0.851

Table 3.3 manifests the values of H before the correction method is applied and after it is applied .while doing so a difference of 0.20 per cent is observes between the values of H for a same time series of data. Showing how Periodogram method could be erroneous for estimating the value of H.

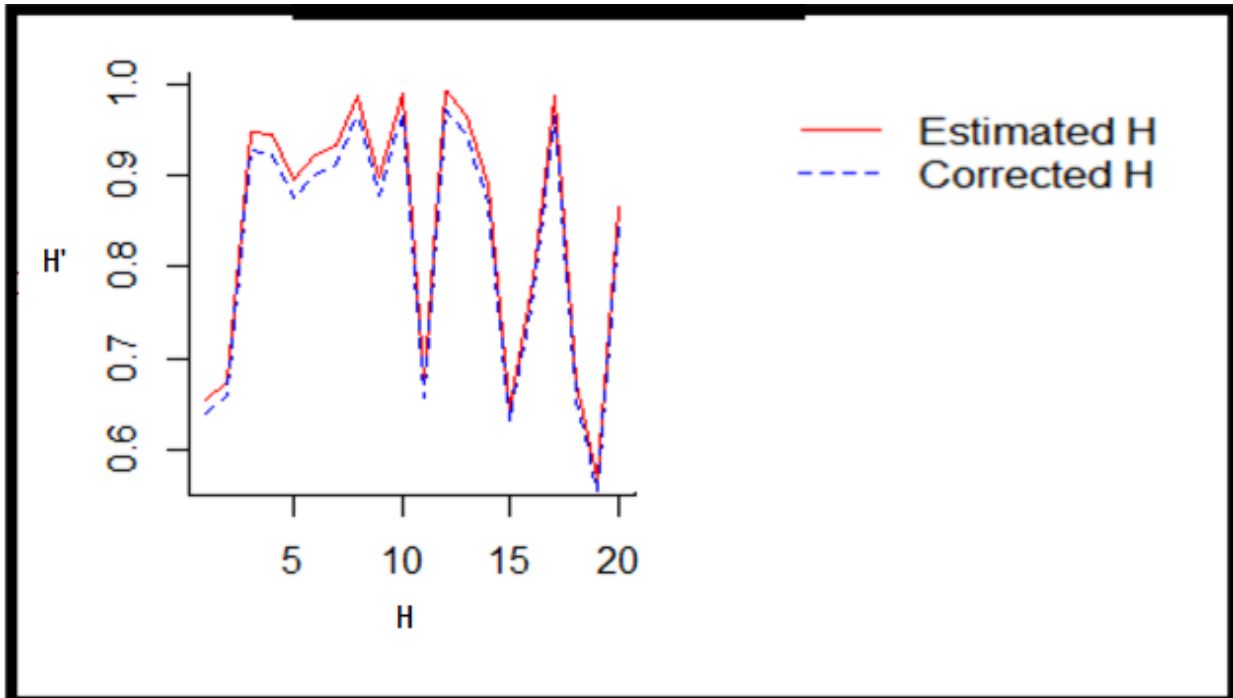


Figure 3.6: Estimated Value of h' by Periodogram and Corrected Value of h After Error calculation by MSE of 0.20 Percent

The value of H in Figure 3.6 is plotted twice one is before passing through MSE calculations and second time after it. The blue lines shows that the value of H after correcting while the red line show the value of H when no error calculation is done. For Periodogram the difference between the two values is around 0.20 per cent.

### 3.6 MEAN SQUARE ERROR AND VARIANCE TIME

The estimation of Hurst parameter by variance time plot method  $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$  and its correction by MSE  $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$  is manifested by Figure 3.7 and table 3.4 shows the corrected and biased Hurst parameter values with an error percentage of 0.20 percent make the performance of this method little biased for the calculation of Hurst parameter as Variance Time method did. Table 3.4 shows the value of H calculated by Variance-Time plot method for the trace files mentioned in chapter 1 in the section 1.5 descriptions of the traffic trace, the files are divided in such a way that twenty series of count data could be extracted and then their respective value of Hurst parameter is calculated.



**Table 3.4. Estimated and Corrected Value of H by Variance Time Method**

No.	Variance Estimated H	Mean Square Error	Corrected H
1	0.681	0.06	0.702
2	0.700	0.07	0.719
3	0.692	0.05	0.622
4	0.643	0.07	0.757
5	0.641	0.07	0.622
6	0.742	0.07	0.757
7	0.643	0.07	0.756
8	0.692	0.05	0.757
9	0.634	0.05	0.757
10	0.858	0.06	0.622
11	0.859	0.08	0.752
12	0.769	0.06	0.573
13	0.843	0.07	0.782
14	0.856	0.07	0.791
15	0.524	0.06	0.455
16	0.763	0.07	0.578
17	0.836	0.07	0.695
18	0.634	0.07	0.791
19	0.986	0.06	0.752
20	0.678	0.07	0.551

Table 3.4 manifests the values of H before the correction method is applied and after it is applied .while doing so a difference of 0.20 per cent is observes between the values of H for a same time series of data. Showing how Variance Time method could be erroneous for estimating the value of H.

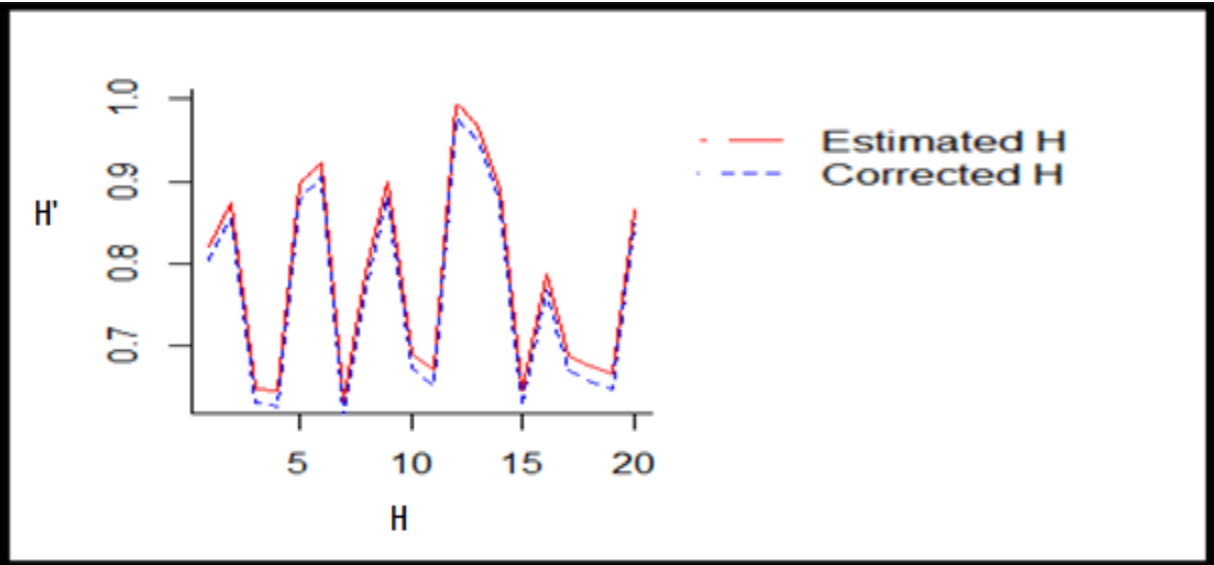


Figure 3.7: Estimated Value of  $h'$  by Variance Time Method and Corrected Value of  $h$  after error calculation by MSE of 0.20 Percent

The value of  $H$  in Figure 3.7 is plotted twice one is before passing through MSE calculations and second time after it. The blue lines shows that the value of  $H$  after correcting while the red line show the value of  $H$  when no error calculation is done. For Variance Time method the difference between the two values is around 0.20 per cent.

### 3.7 SUMMARY OF THE CHAPTER

Hurst Parameters Estimation Techniques are discussed in this chapter. There are twelve well know techniques for finding the Hurst parameter for measuring the presence of self-similarity in the dataset. The twelve methods are R/S method, aggregated variance method, absolute value method, variance of residuals method, Periodogram method and others. The four main methods are discussed in this chapter and their biasedness is measure by Mean Square Error. The best method two methods out of four are proved to be R/S transform and Wavelets as the later has inherit feature of sensing the scaling behaviour as well.

## CHAPTER 4

### INTERNET TRAFFIC ARCHIVE

#### 4.1 INTRODUCTION

As growth of internet users continue to increase manifolds, its structure, underlying dynamics, associated devices, protocols, technology are need to be studied thoroughly for the future dimensions. Analysts and researchers find it hard to get the respective dataset due to legal, social, economic or technical constraints. Resultantly either they drop their projects or they are forced to use the outdated data unable to meet the contemporary need.

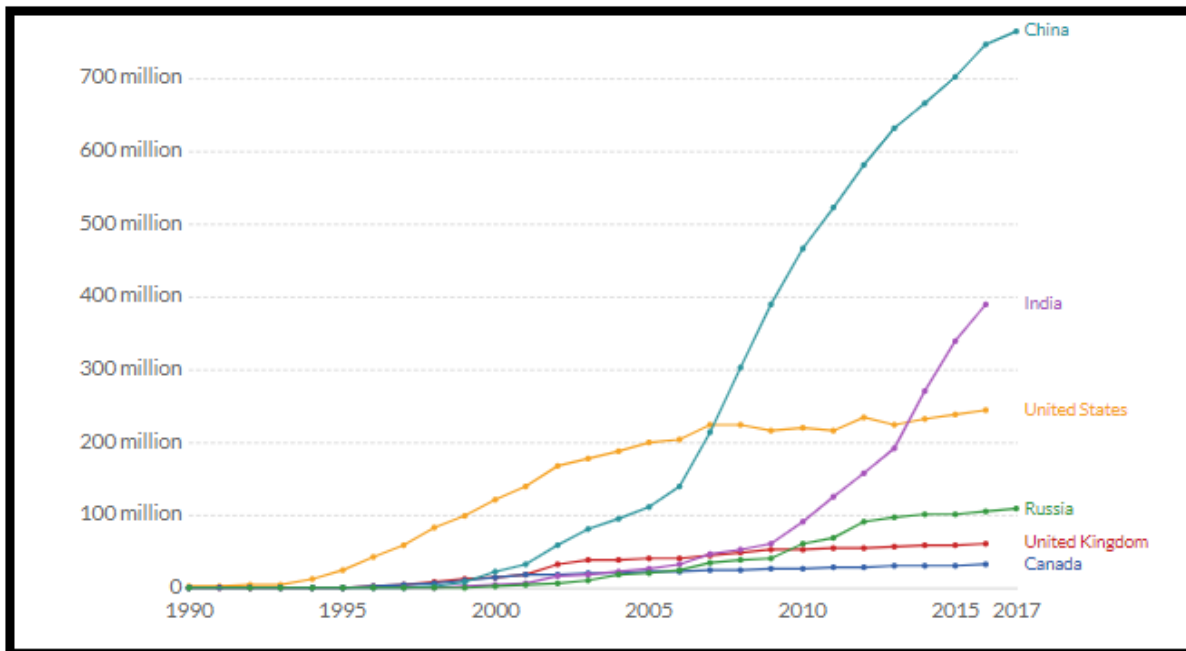


Figure 4.1: Growth of Internet Users in Different Countries from 1960s to 2017<sup>2</sup>

The journey from telephone networks to the outburst of users, manifested by figure 4.1, generating tons of data due to mobile phones, induction of high-speed networks of 3G, 4G and 5G along with increasing applications of Internet of Things (IoT) [1]; has made it necessary to understand the current architecture, scenario, drawbacks and need of internet network to devise a better platform for future internet.

<sup>2</sup> [www.data.worldbank.org/data-catalog/world-development-indicators](http://www.data.worldbank.org/data-catalog/world-development-indicators)

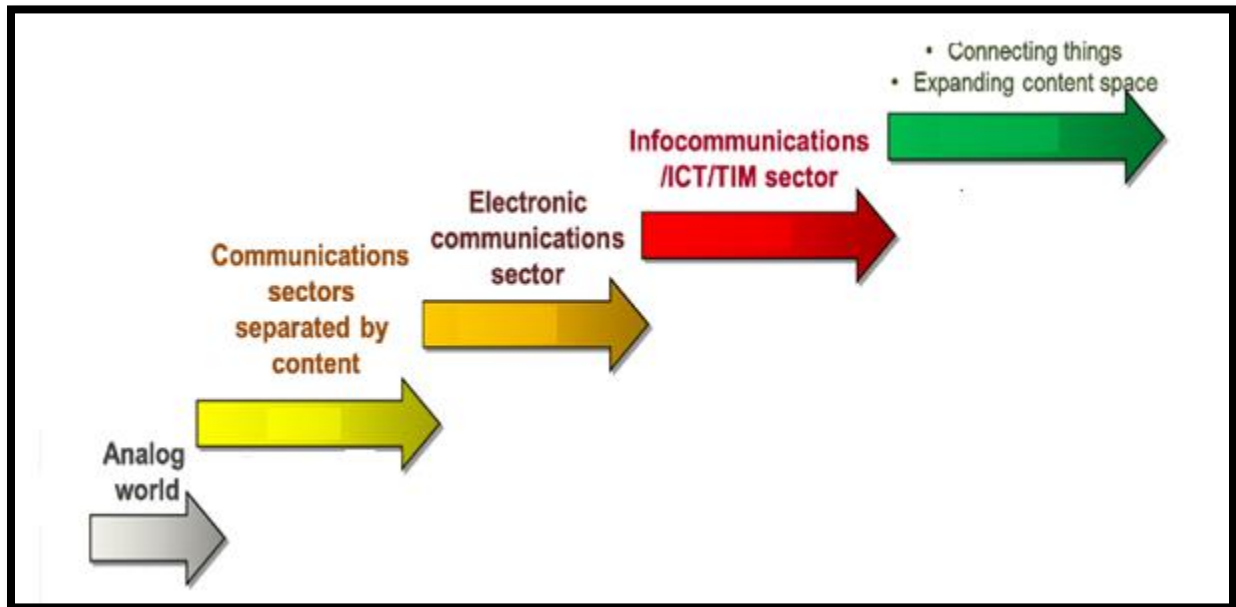


Figure 4.2: Steps of Advancements in the Internet Networks – From Analog World to Digital Universe.

A complete manifestation of different phases of internet networks is accurately depicted by figure 4.2. Many testbeds and traffic archives have been established to comply with prerequisites of research and development (R&D) towards a resilient model of future internet. European Union, United States and South Korea have established are the fore runners in this area.

For instance PlanetLab<sup>3</sup>, OneLab<sup>4</sup>and GLab<sup>5</sup> are testbeds whereas CAIDA<sup>6</sup>, WAND (Wits)<sup>7</sup>, Mawi (WIDE Project)<sup>8</sup> and MOME<sup>9</sup> database are internet traffic archives, helping scientist and researchers in understanding the essentials of internet network for better tomorrow [38]. The ongoing project is an effort to archive the traffic of local Access and Core networks with objective of standing in line with the developed world and play the role in the development of Future Internet.

<sup>3</sup> [www.planet-lab.eu](http://www.planet-lab.eu)

<sup>4</sup> [www.onelab.eu](http://www.onelab.eu)

<sup>5</sup> [www.cosa.th-luebeck.de/en/internet/projects/glab](http://www.cosa.th-luebeck.de/en/internet/projects/glab)

<sup>6</sup> [www.caida.org](http://www.caida.org)

<sup>7</sup> [www.wand.net.nz/wits](http://www.wand.net.nz/wits)

<sup>8</sup> [www.mawi.wide.ad.jp/mawi](http://www.mawi.wide.ad.jp/mawi)

<sup>9</sup> [www.ist-mome.org/database](http://www.ist-mome.org/database)

## **4.2 EXISTING INTERNET TRAFFIC ARCHIVES**

### **4.2.1 THE CENTER FOR APPLIED INTERNET DATA ANALYSIS (CAIDA)**

The Center for Applied Internet Data Analysis (CAIDA) <sup>10</sup> manages network research and builds research infrastructure to support large-scale data collection, curation, and data distribution to the scientific research community [39]. It maintains a growing number of computational and data analysis services. As internet network has reached its prescribed limits CAIDA is helping in routing, security, testbeds management. Its data archive is based upon sampling of internet traffic as opposed to flow or packet archiving. It does not store external data but of its own reaching 32 TB. The storage of meta-data as only and indexing of it help in quick access and does not add any overhead burden on CAIDA servers.

### **4.2.2 WAIKATO INTERNET TRAFFIC STORAGE (WITS)**

The Waikato Internet Traffic (WITS) <sup>11</sup> Storage project maintains and document the internet traffic traces for researchers and scientists. It provides only some traces for public user because of legal constraints. The too it uses is a library Libtrace, written in C language and Java. The library mainly works on packet capturing methods of archiving. This library is used by WITS for multiple types of inputs without any losing any information. The packet captured by it is archived along with its meta-data in the indexing based upon the time and date of capture. However, it is possible to browse the WITS archive using but attempts to download the trace files require IPv6 hosts. Wits has also mirrored its trace on repository. These repositories apply certain restriction for the usage of their data and required to create accounts for that purpose.

### **4.2.3 WIDE PROJECT (MAWI GROUP)**

WIDE (MAWI GROUP) <sup>12</sup> project is an initiative to facilitate researchers in the domain of internet traffic networks. It provides a data repository of backbone traffic. Traffic traces are collected by tcpdump<sup>13</sup> and, after removing privacy information and anonymity, removal of TCP and UDP payloads and IP masking, the traces are made open to the public. Tcprdpriv is used to remove user data while tcpdstat is used to get summary of a tcpdump file in pcap format. It archives packets by tcpdump library. The backbone trace of fifteen minutes is capture and archived on daily basis. The trace is available for public in zipped format. The sampling points are: first one is trans-pacific 1.5Mbps T1 line, from U.S. to Japan link and second is 6Bone is located on a FastEthernet segment connected to NXPIXP-6 (An IPv6 internet exchange point in Tokyo).

---

<sup>10</sup> [www.caida.org](http://www.caida.org)

<sup>11</sup> [www.labs.ripe.net](http://www.labs.ripe.net)

<sup>12</sup> [www.mawi.wide.ad.jp](http://www.mawi.wide.ad.jp)

<sup>13</sup> [www.tcpdump.org](http://www.tcpdump.org)

#### 4.2.4 INTERNET TRAFFIC STATISTICS ARCHIVE (ITSA):

Internet traffic statistics archive works on flow-level traffic measurement by similar to NetFlow<sup>14</sup> or IPFIX<sup>15</sup> from multiple sources. As flow-enabled devices are ubiquitous in networks therefore, flow data is the most suitable for the traffic measurement. It computes pertinent traffic statistics and then uploads those public accessible repositories in the World Wide Web.

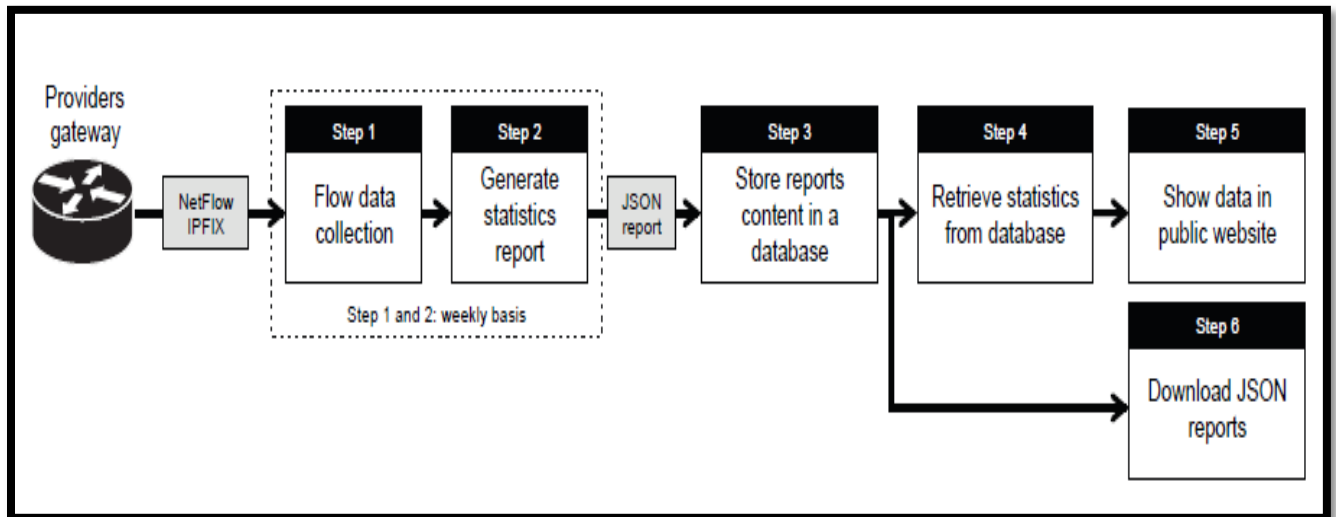


Figure 4.3: The component of ITSA

Figure 4.3 manifesting a complete process of ITSA, from source of traffic to reports upon on internet after statistical analysis. The archive begin its route from the router the forwards the traffic towards archive which afterwards captured by NetFlow – a tool for flow capture. Afterwards that captured data is processed and repots through JSON are made.

#### 4.3 METHODS OF INTERNET TRAFFIC ARCHIVING

Network traces helps researchers, analysts, scientist and an organization by providing actionable information for network monitoring, management and decision making effectively. The whole process of internet traffic capturing, archiving, and evaluating is colossal and critical. With high-speed networks and enormous network traffic volumes, full-packet traffic capture requires countless bytes of storage for a single day. So the possibility of packet capture depends upon the vast resources and resilient capabilities.

14 B.Claise, "Cisco Systems NetFlow Services Export Version 9" RFC 3954, 2004.

15 J.Quittek, T.Zseby, B.Claise, and S. Zander, "Requirements for IP Flow Information Export (IPFIX), "RFC 3917, 2004.

On the other hand, layered network storage with information of only critical or effective types of traffic in full-packet captures and residual as summary data, can help in resolving the overhead issues while providing the detailed information. This chapter provides a brief insight for the potent methods of traffic archiving, summarized by Table 4.2.

```

> Frame 48: 213 bytes on wire (1704 bits), 213 bytes captured (1704 bits) on interface 0
v Ethernet II, Src: IntelCor_e7:f4:23 (30:3a:64:e7:f4:23), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  > Destination: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  > Source: IntelCor_e7:f4:23 (30:3a:64:e7:f4:23)
  Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 192.168.1.102, Dst: 239.255.255.250
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 199
  Identification: 0x5305 (21253)
  > Flags: 0x4000, Don't fragment
  Time to live: 1
  Protocol: UDP (17)
  Header checksum: 0x7418 [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.1.102
  Destination: 239.255.255.250
v User Datagram Protocol, Src Port: 60419, Dst Port: 1900
  Source Port: 60419
  Destination Port: 1900
  Length: 179
  Checksum: 0x8d27 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 5]
> Simple Service Discovery Protocol

```

Figure 4.4: Information of Packet Captured by Wireshark<sup>16</sup>

Figure 4.4 manifest information of a packet captured by Wireshark. It shows the frame number in a trace, source IP, destination IP, source port and destination port and other important information.

### 4.3.1 FULL PACKET CAPTURE

Full packet capture or full content collection exhibit full packet capture of pcap. In this method full trace is capture that passes a captured point. In short it collects processes and stores every packet that traverses through network for later use. It contains full header and payload information. Figure 4.4 showing the packet capture information by Wireshark - packet capturing tool. This stores information in pcap file of packets such as the frame number, interface, timestamp, epoch, protocols used, source and destination port numbers, source and destination IPs, payload and other information. Though this method provides an include mirror of traffic but on the cost of resources for storage, processing and analysis.

<sup>16</sup> [www.wireshark.org](http://www.wireshark.org)

So, in the worst case scenario the amount of information stored by a packet capture method would be equal to:

$$\text{Information} = \text{Capacity} \times \text{Time} \quad (4.1)$$

Here, capacity is total bandwidth and time is the captured time. In the normal case the amount of information would be:

$$\text{Utilization} \times \text{Time} \quad (4.2)$$

Here, utilization average percentage of bandwidth used out of total available bandwidth.

#### 4.3.2 NETWORK FLOW CAPTURE

Another method of capturing the internet traffic is to store the network flow. According to RFC 3917, requirements for IP Flow Information Export (IPFIX) [12], a flow is a set of IP packets passing through a point in certain interval of time. Every packet of a flow has some common properties. Such as:

- Packet header field, transport header field, or application header field.
- Fields derived from packet processing. For instance next hop IP address, the output interface, etc.

One can put it in another way that flow represents streams of network packets by generating one flow record for all captured packets that are lay within the same time frame , contain same port addresses, and use same protocol. Flow recorded file also contain the total number of packets, the time elapse between start and end and the flags within the headers of different packets. Flow does not store packet payload or much header information.it requires minimum storage and processing resources but at the cost of incomplete analysis with meager sets of information. Flow can be stored unidirectional or directionally. Table 4.1 describes the pros and con of both approaches. However, the worst case in terms of network flow would be:

$$\text{Information} = \text{Size} \times \text{Max Flows} \times \text{Time} \quad (4.3)$$

Here size is the size of flow, max flow is the maximum possible flow per time period, and time is the time for the capture of the flow. In normal case the amount of information that stored would be:

$$\text{Information} = \text{Size} \times \text{Average Flows} \times \text{Time} \quad (4.4)$$

Here Average flow is the average number of flows per time period.



**Table 4.1. Comparison of Unidirectional and Bidirectional Flow**

Considerations	Unidirectional	Bidirectional
Association of flow to determine Conversations / Sessions	Possible only manually	Yes
Wrong Association of two flows in conversation /session	Manual association is possible	Yes
Identification of Packet sizes between two flows	Yes	Only if captured and stored with the flow
Identification of who started conversation / session	No	Only if captured and stored with the flow

In table 4.1. Comparison of Unidirectional and Bidirectional Flow is summarized. The direction of flow plays important role in assessing the properties of network performance, bandwidth, queuing performance and others.

### 4.3.3 AUGMENTED FLOW CAPTURE

Augmented Flow is something in between the packet capture and the basic network flow. In this procedure more information from packet header and payload is added to the network flow information such as application labels, entropy, operating system fingerprints, etc. It may contain additional information such as location of source and destinations.

It also contains metadata that is network packets information that could be decoupled from the flow. The worst case information stored could be:

$$\text{Information} = \text{Size} \times \text{Max Flows} \times \text{Time} \quad (4.5)$$

The normal case information would be:

$$\text{Information} = (\text{Size} + \text{Metadata size}) \times \text{Average Flows} \times \text{Time} \quad (4.6)$$

Table 4.2. Types of Capturing Network Traces for Internet Traffic Archiving

Types of Capture	Storage Component	Information provided	Storage& overhead
Full Packet Capture	Entire packet	Most information	High storage and overhead
Augmented Network Flow	Network flow , packet fields and external data	Moderate information	Moderate to high storage and overhead
Network Flow	Packet header	Least information	Least storage and overhead

In table 4.2 different types of capturing methods are compared. It is clear from the comparison that if more information is required than Full Packet capture is best method otherwise Network Flow is best to use because of least storage resources required.

#### 4.4 INDEXING OF TRACES

After capture of traces indexing is one of the core areas of archiving. As it helps in quick access of the required data. It optimizes the performance of archive databases by minimizing the number of storage accesses as a result of a query processing. In short, an index or database index is a data structure which is used to quickly locate and access the data. Traditionally binary tree algorithm has been used for the indexing. Current a more efficient method of KEY STORE is used for quick access. Information index must have create index for massive data, high speed data processing, a data structure ,high repeatability and change of index must be instant. Bit map indexing and the indexing based on Hurst parameter is defined briefly below.

##### 4.4.1 BITMAP INDEXING

The core concept of bitmap indexes is to use one bitmap for every distinct value. It is possible to reduce the number of bitmaps used by using different encoding methods. For instance, it is possible to encode A

distinct values using  $\log(A)$  bitmaps with binary encoding. However, finding the optimal encoding method that has efficient query performance, small index size and resilient index maintenance still remains a challenge. This technique is used for huge databases, when column is of low cardinality and they are most frequently used. Bitmap Indexing use a bit vector for indexing. It uses bit logical operations of AND, OR, NOT or XOR to process queries. It is a scientific index used by extremely huge data without any amount of change. Compression and encoding are the main part of algorithm of bitmap indexing.

#### **4.4.1.1 COMPRESSION**

Compression is done to save storage space. A lot of work has been done on this subject [15][16]. Bitmap compression are : Byte-aligned Bitmap Code, the Word-Aligned Hybrid code, the Partitioned Word-Aligned Hybrid (PWAH) compression,[17] the Position List Word Aligned Hybrid,[18] the Compressed Adaptive Index (COMPAX),[19] Enhanced Word-Aligned Hybrid (EWAH) [20]. Theses algorithms required a minute effort to compress and decompress. They can also directly take part in bitwise operations without decompression. This gives them edge over traditional compression.

#### **4.1.1.2 ENCODING**

Encoding means each distinct value has one encoded value. The encoded values can be reduced by applying some function. Either by log or any other blackbox function to map the value and the indexed code. Table 4.3 is an example of encoding for bitmap indexing. Here the first column represent the row id, while second column is actual data and third and fourth column are yes and no . The criteria for Self-Similarity are based upon the value of Hurst parameter, as described in the early chapters. If value of it is greater than 0.5 than it would be yes or 1 otherwise it would be no or 0.Theses encoded bits are than compressed by a function to take less pace and identify each row correctly.

#### **4.4.2 H- BASED INDEXING**

Hurst parameter is a well know measure of self-similarity. Index based on the value of H would manifest the how long a tailed decay would be and how persistent the scaling function in the traces. This would help in understanding the patterns of burstiness introduced by different layers of TCP/IP protocols and OSI layers. For instance the traffic brustiness of Application layer due to different applications and their protocols – WhatsAppVoice, FacebookCall, Skype, etc. These effects would be discussed in the following chapters. The husrt parameter value is computed here by wavelet based algorithm, as described in Chapter 2.

Traces in Traffic Repository		
Show <input type="text" value="10"/> entries		
Hurst Value (H)	Traffic Traces	trace files Size
0.8642204	wifi_02_20191501	2576
0.8404487	wifi_06_20191501	256
0.7233553	wifi_05_20191501	5648
0.6671554	wifi_03_20191501	432
0.6558353	wifi_01_20191501	2576

Showing 1 to 1 of 1 entries

Figure 4.5: A Hurst Parameter Based Index of Trace Files

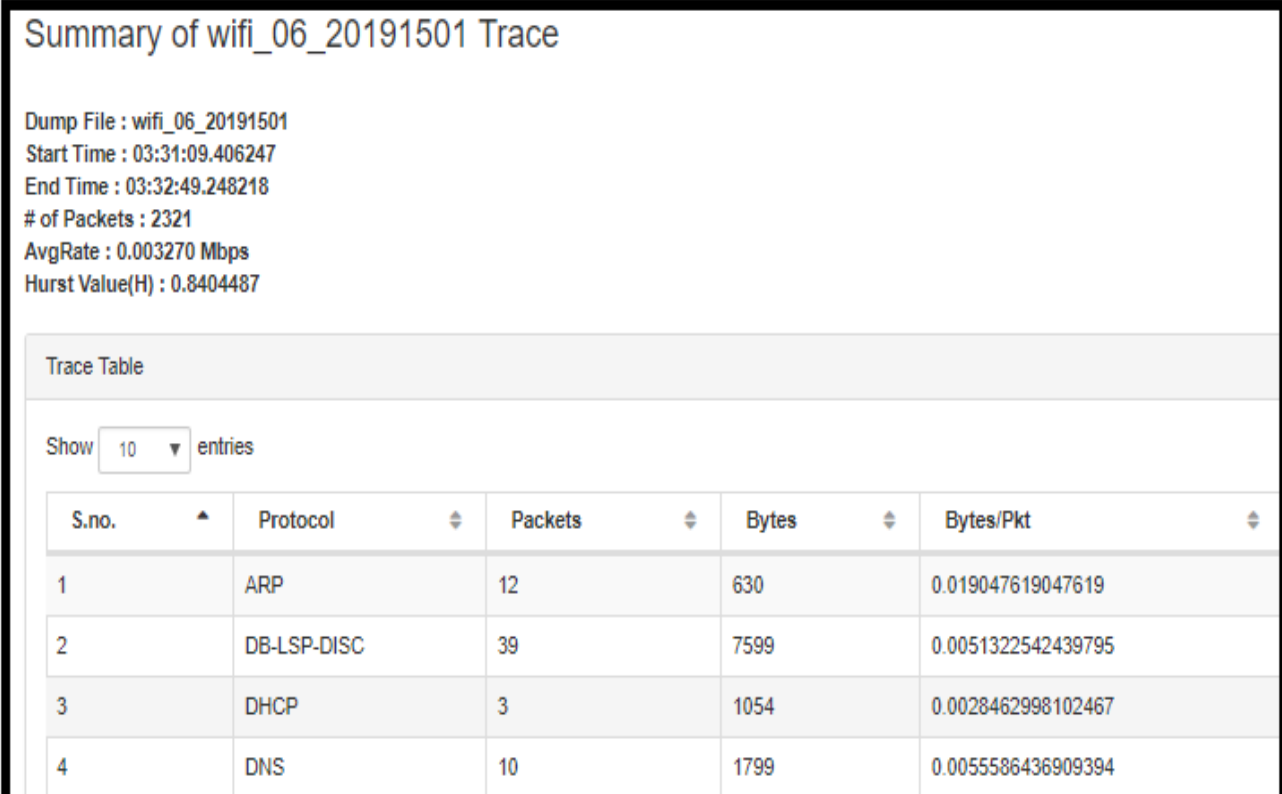
Figure 4.5 manifesting the traces of file on website of varying size ranging from few KBs to MBs, the traces are of Wi-Fi and LAN. In the column for value of Hurst Parameter different value of H is presented ranging from 0.86 to 0.65. One important aspect needed to be mentioned here is that value of H does not depend the file size only as for same file size it could be same or different as it is more depend upon the pattern of variance and autocorrelation in the series of count data for a particular trace file.

#### 4.5 NRPV – INTERNET TRAFFIC ARCHIVE

The inspiration of internet traffic archive originated from the growing demand of internet technology which is saturated to its maximum bounds. A universal model for future is possible only when current dynamics of network is studied, examine and analyzed for both Access and Core networks. This will be the first Internet traffic archive in Pakistan which will be accessible and citable word wide. These archives are available in New Zealand, Japan and USA only at the moment. This project will also develop an Internet traffic archive with accurate time stamping and obeying user privacy and service level agreement laws. Hence this project will establish Pakistan's role in international Future Internet initiatives.

#### 4.5.1 DESCRIPTION OF TRACES

The traces are captured by Wireshark and based on packet capturing method of archive. The duration of traces vary from 15 minutes to 30 minutes. They can be captured by a routine on daily basis for the required time or may be captured manually. The traces are of Ethernet and Wi-Fi networks of NED University. These are processes and anonymized first to upload on the archived as shown in figure 4.6. The size of traces varies from MBs to GBs.



The screenshot displays the 'Summary of wifi\_06\_20191501 Trace' interface. It includes the following summary statistics:

- Dump File : wifi\_06\_20191501
- Start Time : 03:31:09.406247
- End Time : 03:32:49.248218
- # of Packets : 2321
- AvgRate : 0.003270 Mbps
- Hurst Value(H) : 0.8404487

Below the statistics is a 'Trace Table' with a 'Show 10 entries' dropdown. The table contains the following data:

S.no.	Protocol	Packets	Bytes	Bytes/Pkt
1	ARP	12	630	0.019047619047619
2	DB-LSP-DISC	39	7599	0.0051322542439795
3	DHCP	3	1054	0.0028462998102467
4	DNS	10	1799	0.0055586436909394

Figure 4.6: A Trace Files along with the summary of Trace

This figure 4.6 represent the summary of a trace file represent the start time, end time, number of packets and the protocols in that selected trace.

#### 4.5.2 GRAPHICAL PRESENTATION OF TRACES

The trace files after uploaded on the archived website are manipulated to show their contents in the form of Graph based on the philosophy: a picture worth thousands words, they are described below separately.

#### 4.5.2.1 PROTOCOLS DISTRIBUTION OF A TRACE FILE

The protocol distribution is manifested by bar chart in which on x-axis the protocols in any given trace is listed such as ARP, DHCP, ICMP, IGMP, MSDN, TCP, TLS, UDP, etc. While on y-axis the number of packets of respective protocols is given.

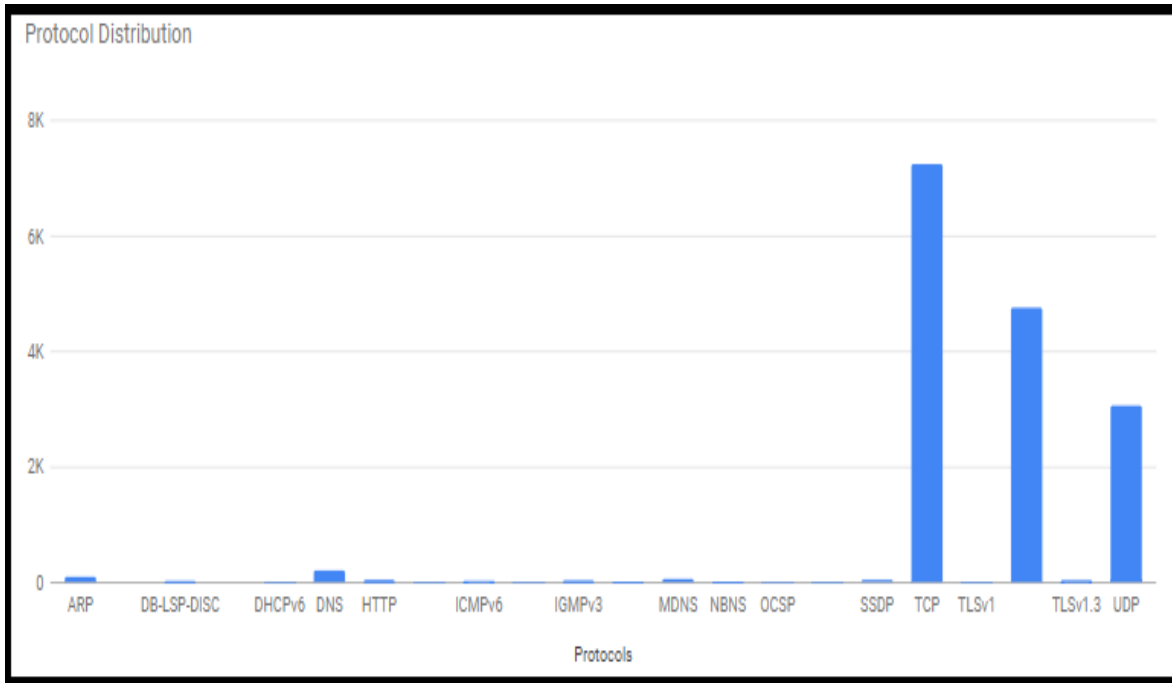


Figure 4.7: Bar Chart - Protocol Distribution of a Trace

Figure 4.7 is of bar chart for a trace file depicting protocol distribution that is of ARP, HTTP, and ICMPv6, TCP and UDP and others.

#### 4.5.2.2 PERCENTAGE OF PACKETS PER PROTOCOL

A pie chart is used to represent the percentage of packets of each protocol. This approach gives a holistic picture of the protocol present in the trace file. The dominant protocol finds a large part for them in pie whereas the small number of packets of protocol finds the space as their meager presences. Protocol layers can consist of packets that have not any higher layer protocol, so the aggregate of all higher layer packets could not equal to the protocols packet count. For instance, if TCP packets dominate 98.5% of a capture but the sum of the sub protocols such as TLS, HTTP, etc. is much less. The reasons for this behaviour are plenty: may be caused by continuation frames, TCP protocol overhead and other undissected data. Whereas, figure 4.8 describe a packet distribution for protocols in term of their percentages and showing the dominant protocol traffic in more efficient way. The legends on the right hand side labeled the manifest protocols in pie chart.

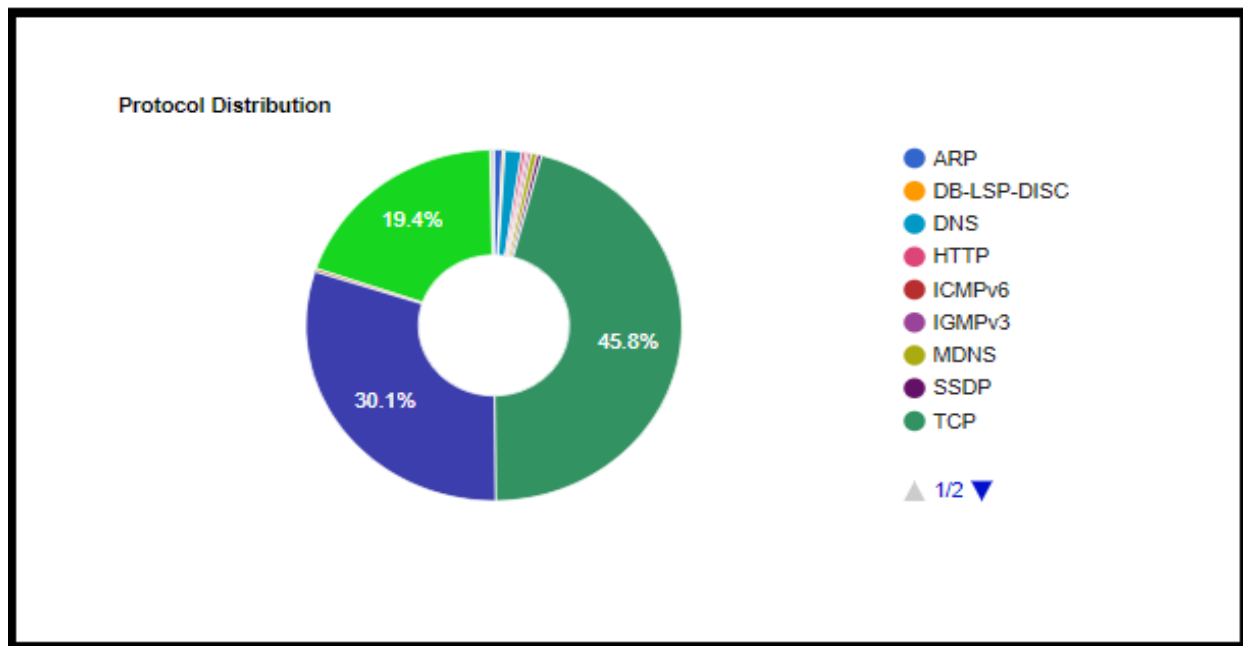


Figure 4.8: Percentages of Packets for each protocol.

The pie chart in figure 4.8 represents the protocol distribution of a selected trace file. The highest number of packets of a protocol is of TCP that is of 45.8 per cent, second is of ARP 30.1 percent and third highest is of UDP comprised of 19.4 percent.

#### 4.5.2.3 PACKET COUNT IN A TRACE FILE

Networks communication is based upon small chunks called packets. A trace file contains hundreds of packets depend upon the characteristic of the network from which the trace is capture. Along with the time of capture that is how long a trace is captured. Packet count is the number of packets in a trace file. It is of importance because sending more packets per second creates problems for network connections, resulting in packet loss, lag spikes and ultimately a complete blackout altogether. In one experiment, decreasing the average send rate in a full from 200 packets per second to 200 has reduced the connection errors by around 25%. This is big number in statistical analysis of network performance and resource allocations and would have significant impact on the network as well. There are many ways to show the packet count of a capture but the graphical representation is easy to manifest. Figure 4.9 depicts the packet count for a trace file against the time of its capture.

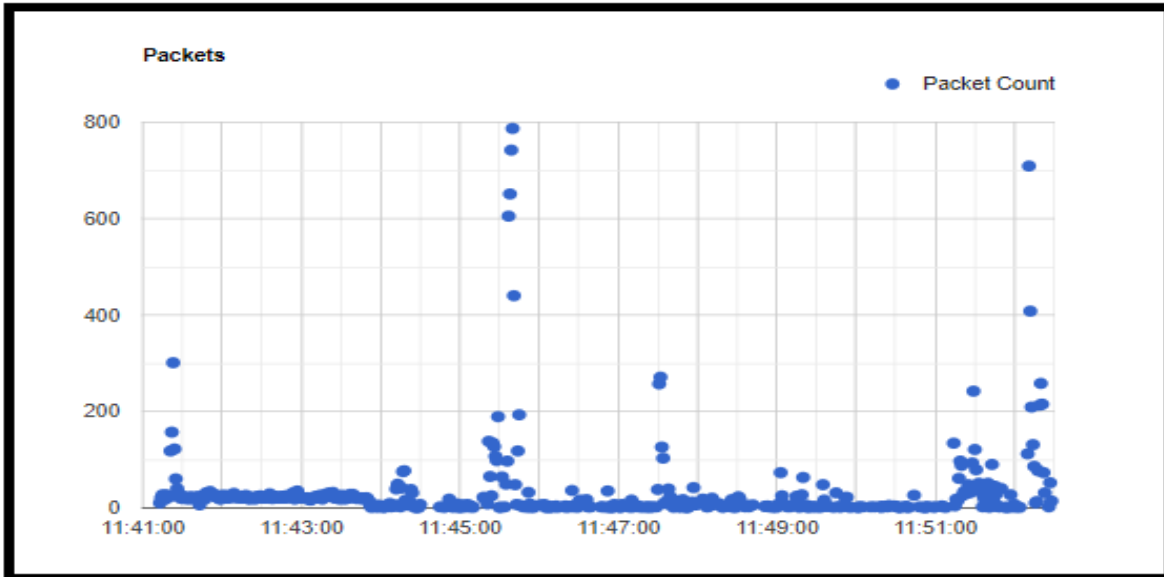


Figure 4.9: Packet Count of a Trace File

The scattered plot t in figure 4.9 represents the packet distribution over time scale, the trace in figure is of 10 minutes and the number of incoming packets is ranging from tens from hundreds.

#### 4.6 SUMMARY OF THE CHAPTER

Internet traffic archive is one of basic demands of today’s growing high-speed network for better understanding of it. It is an aid for scientists, researchers and student who are trying to understand the underlying dynamics of contemporary 3G, 4G, 5G, ubiquitous and other networks and their impacts [37]. This is necessary to outline the demand of Future Internet as the growth of users of such technologies is increasingly creating constraints on the performance and resources of the networks. Internet traffic archiving has been undertaken by United States, South Korea, European Union, New Zealand, Japan and others as mentioned in the chapter. The traffic from both access and core networks are collected from the networks and after processing and analysis they are indexed before uploading on a website for public access [39]. Though there are many social, political, legal, technological and economic constraints in accessing them freely but there are some archives that provide free of cost traces for analysis and processing. The archive for this project would be a public archive and would have both traffic of access and core networks. The indexed currently used is of Hurst parameter but it would be indexed by Bitmap algorithm for faster accessing and robust performance in minimal resources. Further graphical representation after processing of data is one of the characteristic features of this archive. They manifest the protocol distributions, packet count, and bandwidth and so on.



## CHAPTER 5

### APPLICATION LAYER TRAFFIC AND SELF-SIMILARITY

#### 5.1 INTRODUCTION

Will Leland, Walter Willinger et al [7] have discovered self-similarity in Ethernet LAN traffic of data-link layer that is layer 2 of the Open Systems Interconnection (OSI) model. Then layer 3 was studied for the same purpose. However, few studies have been conducted to find the dominant factor responsible for generating self-similar network traffic. In this chapter the impact of application layer traffic on over all internet traffic (TCP flow and its aggregate) would be examined. Usually in congested networks Transport layer traffic dominate Application layer traffic for stimulating the burstiness in traffic. But with ubiquitous computing, mobile networks and growing applications of social media with outburst of users the trend seems to be changed in the near future. By studying the nature of application layer traffic QoS could be improved, performance of overall system could be enhanced and future implications of high-speed ubiquitous computing could be met. The methods for finding self-similarity as described in chapter 3 would be used to find the dominating protocol of Application layer in inducing self-similarity in whole TCP flow.

#### 5.2 INTERNET TRAFFIC PROTOCOL BREAKDOWN

protocol	packets	bytes	bytes/pk
total	88815145 (100.00%)	56335043211 (100.00%)	634.30
ip	84845860 ( 95.53%)	52605595128 ( 93.38%)	620.01
tcp	50235480 ( 56.56%)	43192350051 ( 76.67%)	859.80
http	15786444 ( 17.77%)	17401216367 ( 30.89%)	1102.29
https	18828348 ( 21.20%)	19915160422 ( 35.35%)	1057.72
smtp	168509 ( 0.19%)	55182053 ( 0.10%)	327.47
ftp	227006 ( 0.26%)	13787963 ( 0.02%)	60.74
ssh	3044687 ( 3.43%)	986786274 ( 1.75%)	324.10
dns	15493 ( 0.02%)	1512441 ( 0.00%)	97.62
bgp	2815 ( 0.00%)	898598 ( 0.00%)	319.22
other	12162178 ( 13.69%)	4817805933 ( 8.55%)	396.13
udp	7714012 ( 8.69%)	5606319142 ( 9.95%)	726.77
dns	487825 ( 0.55%)	169630222 ( 0.30%)	347.73
https	3694752 ( 4.16%)	4765617644 ( 8.46%)	1289.83
other	3531417 ( 3.98%)	671066064 ( 1.19%)	190.03
icmp	22819526 ( 25.69%)	1395159213 ( 2.48%)	61.14
gre	4071526 ( 4.58%)	2407955958 ( 4.27%)	591.41
ipsec	5295 ( 0.01%)	3808370 ( 0.01%)	719.24
ip6	21 ( 0.00%)	2394 ( 0.00%)	114.00
frag	67565 ( 0.08%)	98097377 ( 0.17%)	1451.90

Figure 5.1: Percentages of Packets for each Protocol.

Figure 5.1 manifesting the breakdown of protocols of various layers such as application layer, transport layer, network layer and others. The dominant protocol in internet TCP/IP architecture is usually TCP protocol confirmed by the figure after which UDP, ICMP, GRE, IPsec and others. One of the measure sections of TCP protocol is http and https. Hyper Text Transfer Protocol Secure https is the secure version

of HTTP. It is used for secured communications and it employed various encryption algorithms. The application layer traffic is usually of https, used by various modern day applications. In order to find the nature of traffic and their self-similar behaviour they must be studied.

### 5.3 APPLICATION LAYER DEEP PACKET INSPECTION

In order to look deeper into the traffic trace deep packet inspection is mandatory. This facility is provided by Deep packet Inspection<sup>17</sup> (dpi) library by ntop<sup>18</sup>. This library provides one of the techniques of identifying the protocols embedded in packets without associated any particular port number with them.

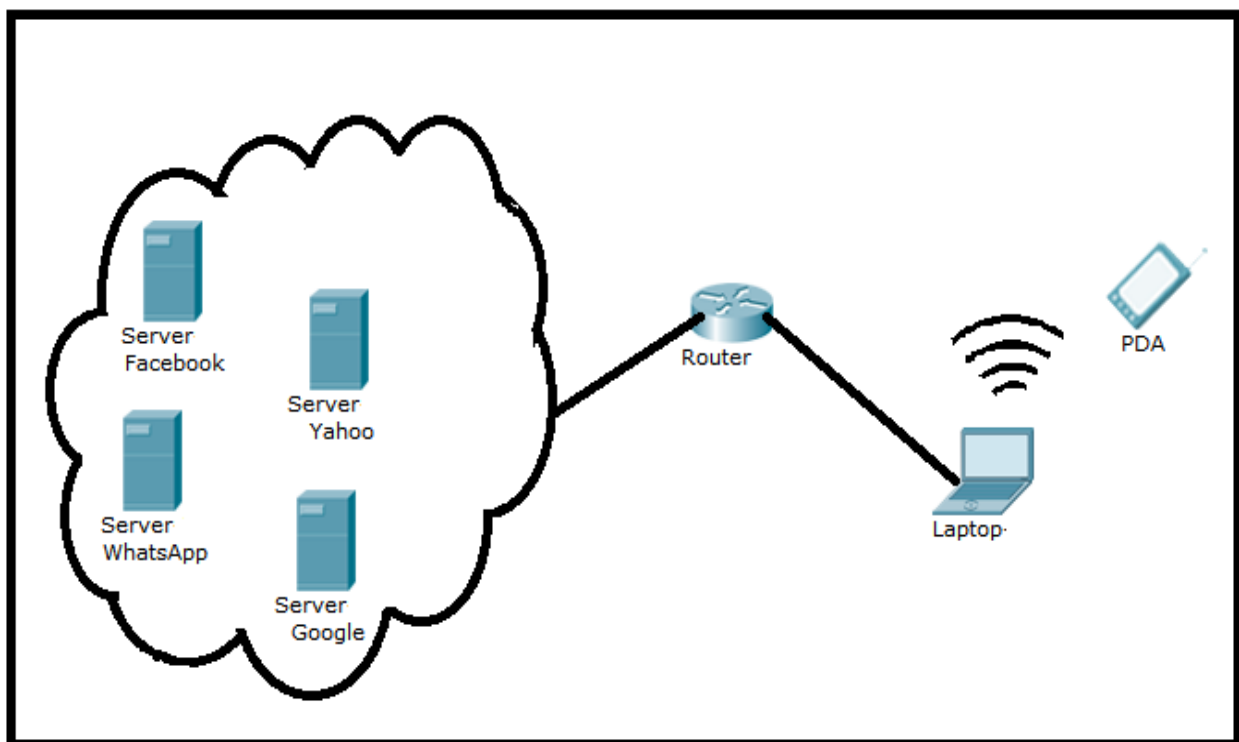


Figure 5.2: Experimental Setup for Traffic Trace Capture

Figure 5.2 manifesting the experimental setup for the packet capture of various Applications. The data travels across the network back and forth from the ubiquitous devices to the servers. The data is captured by the Wireshark installed in the machine in the network.

<sup>17</sup> <https://www.ntop.org/products/deep-packet-inspection/ndpi/>

<sup>18</sup> [www.ntop.org](http://www.ntop.org)

1263	1223.596588	192.168.1.102	31.13.75.49	STUN
1263	1223.599953	192.168.1.101	192.168.1.102	ICMP
1263	1223.679875	157.240.24.52	192.168.1.102	WhatsAppVoice
1263	1223.679894	157.240.24.52	192.168.1.102	WhatsAppVoice
1263	1223.932561	192.168.1.102	157.240.7.35	SSL.Facebook
1263	1223.934409	192.168.1.102	157.240.7.35	SSL.Facebook
1263	1224.020983	157.240.7.35	192.168.1.102	SSL.Facebook
1263	1224.021076	192.168.1.102	157.240.7.35	SSL.Facebook
1263	1224.021160	157.240.7.35	192.168.1.102	SSL.Facebook
1263	1224.021180	192.168.1.102	157.240.7.35	SSL.Facebook
1263	1224.280715	157.240.7.35	192.168.1.102	SSL.Facebook
1263	1224.280720	192.168.1.102	157.240.7.35	SSL.Facebook
1263	1224.280741	157.240.7.35	192.168.1.102	SSL.Facebook
1263	1224.288743	192.168.1.102	157.240.7.35	SSL.Facebook
1263	1224.925422	192.168.1.102	157.240.7.20	SSL.Facebook
1263	1225.013288	157.240.7.20	192.168.1.102	SSL.Facebook
1263	1225.292898	157.240.7.20	192.168.1.102	SSL.Facebook
1263	1225.293000	192.168.1.102	157.240.7.20	SSL.Facebook
1263	1225.486041	192.168.1.101	224.0.0.251	MDNS
1263	1225.719884	192.168.1.102	157.240.7.54	SSL.WhatsApp
1263	1225.719903	192.168.1.102	157.240.7.54	SSL.WhatsApp
1263	1225.925010	192.168.1.102	157.240.7.20	SSL.Facebook
1263	1226.010026	157.240.7.20	192.168.1.102	SSL.Facebook
1263	1226.305520	157.240.7.20	192.168.1.102	SSL.Facebook
1263	1226.305615	192.168.1.102	157.240.7.20	SSL.Facebook
1263	1226.646534	192.168.1.102	157.240.7.53	SSL.Facebook
1263	1226.735613	157.240.7.53	192.168.1.102	SSL.Facebook
1263	1226.735680	192.168.1.102	157.240.7.53	SSL.Facebook
1263	1227.033822	192.168.1.102	157.240.7.03	SSL.Facebook
1263	1227.167894	157.240.7.53	192.168.1.102	SSL.Facebook
1263	1227.167915	157.240.7.53	192.168.1.102	SSL.Facebook
1263	1227.167934	192.168.1.102	157.240.7.53	SSL.Facebook
1263	1227.167968	157.240.7.53	192.168.1.102	SSL.Facebook
1263	1227.167976	192.168.1.102	157.240.7.53	SSL.Facebook

Figure 5.3: A Traffic Trace Capture by Deep Packet Inspection

Deep packet inspection read the trace and decrypts the https packets to find the protocols they have been using as Figure 5.3 is showing a trace of internet traffic with variety of https protocols.

#### 5.4 AGGREGATED TRAFFIC & SELF-SIMILARITY

A network trace of 132,099 packets is captured through Wireshark<sup>19</sup> software is then plots using Wavelet transform method to estimate the value of H-self-similarity parameter.

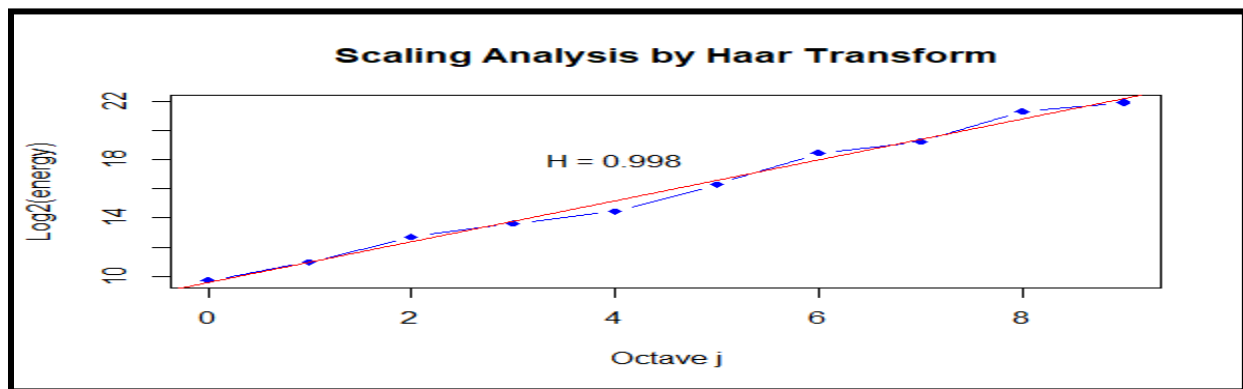


Figure 5.4: Measure of Self-Similarity  $h = 0.989$  by Wavelet Scaling Analysis Method for Aggregated Traffic Trace of 15 minutes Composed of 132,099 Packets

Figure 5.4 is of Wavelet Transform; energy vs. octave  $j$  plot; here  $h$  is calculated by regression fit which gives a slope  $\beta$  to calculate  $h = 0.998$ . According to analysis of chapter two the best estimators for Hurst

<sup>19</sup> www.wireshark.org

parameters are proved to be Wavelet based method and R/S transform. The aim of performing the analysis by all four estimators is to make sure that self-similarity is persistent in the dataset no matter what techniques is employed to find it.

## 5.5 ANALYSIS OF WHATSAPP VOICE, FACEBOOK, GOOGLE TRAFFIC

With the growth of internet users and reciprocal variety of https applications the behaviour of internet traffic is claimed to be changed. As it is thought that OSI layer seventh traffic is making the aggregated traffic more self-similar than the TCP was before. The ongoing section of the report would examine this claim by measuring the self-similarity with and without the traffic induced by individual applications using the R/S transform method and then conclusions would be drawn.

### 5.5.1 SELF - SIMILARITY INDUCED BY DIFFERENT BY HTTPS PROTOCOLS

#### 5.5.1.1 WHATSAPP & SELF-SIMILARITY

WhatsApp Messenger is an Android Application also works on IOS, provides cross-platform messaging and Voice over IP (VoIP) service. The allows the sending of text messages and voice calls, as well as video calls, images and other media, documents, and user location. The value of H given by the various algorithms of h is highly self-similar. For this purpose the analysis is performed on 800 packets generated by SSL.WhatsApp protocol working under the cover of https.

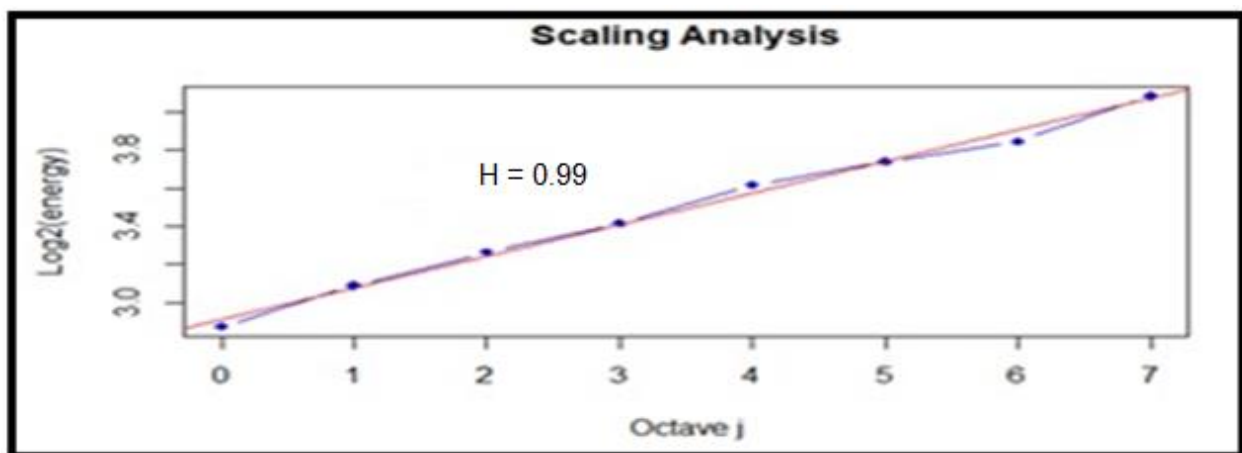


Figure 5.5: Measure of Self-Similarity  $H = 0.99$  by Wavelet Transform for WhatsApp packets for 800

In figure 5.5 Wavelet's  $H = 0.99$  As h is closer to 1 the chance of self-similarity and hyperbolic decay following power law is obvious. So the data packets for whatsapp are highly self-similar and bursty.

### 5.5.1.2 WHATSAPP VOICE & SELF-SIMILARITY

WhatsApp voice calls uses protocol WhatsAppVoice under the cover of https unfolded by dpi. The packets for the analysis of self-similarity are of 1200 captured after making various whatsapp calls.

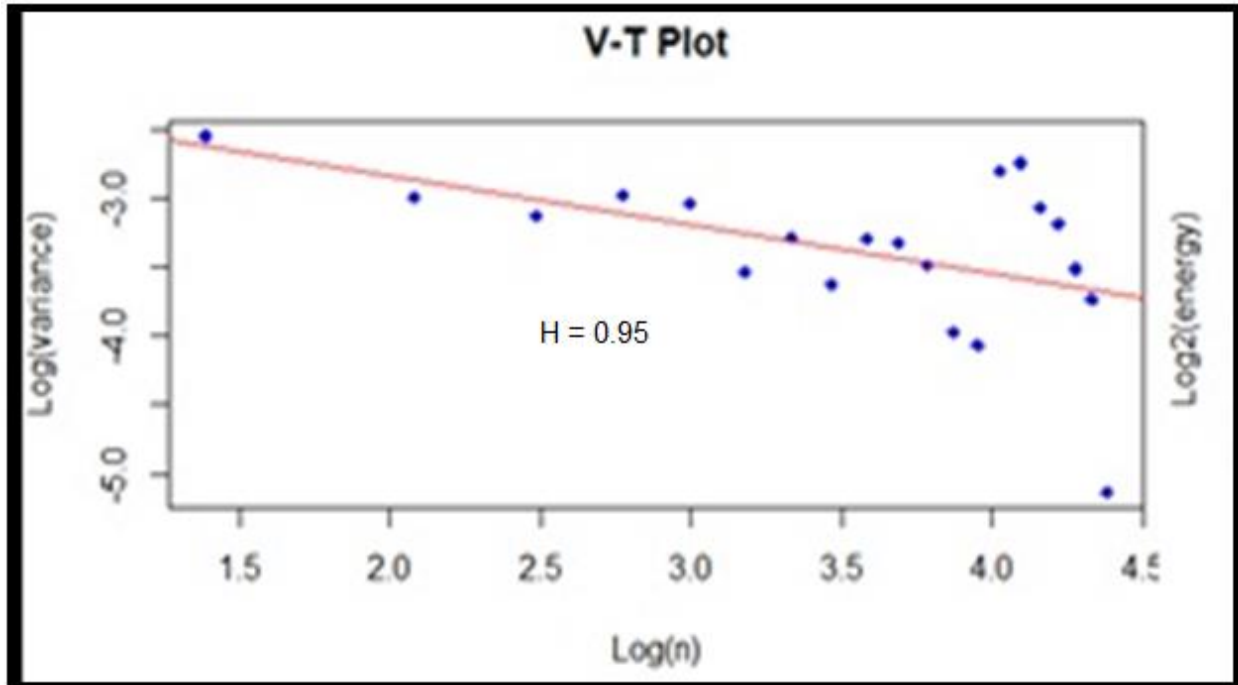


Figure 5.6: Measure of Self-similarity  $h=0.95$  by V-T for WhatsApp Voice packets for 1200

Self - similarity induced by these packets are also strong as manifested by figure 5.6.  $H$  is 0.95 according to Wavelet Transform. The impact of whatsapp traffic over the entire aggregated traffic stream would be studied later.

### 5.5.1.3 FACEBOOK & SELF-SIMILARITY

Facebook employs secured channel for communication that is https. It gives messaging, voice calls, video calls. It users are 1.74 billion across the world. With such high number of user the behaviour of its traffic is important to analyze for the better management of networks and its performance. Its heavy tail behaviour is studied in this section. The decay of Facebook time series packet is also hyperbolic decay and obeys power law.

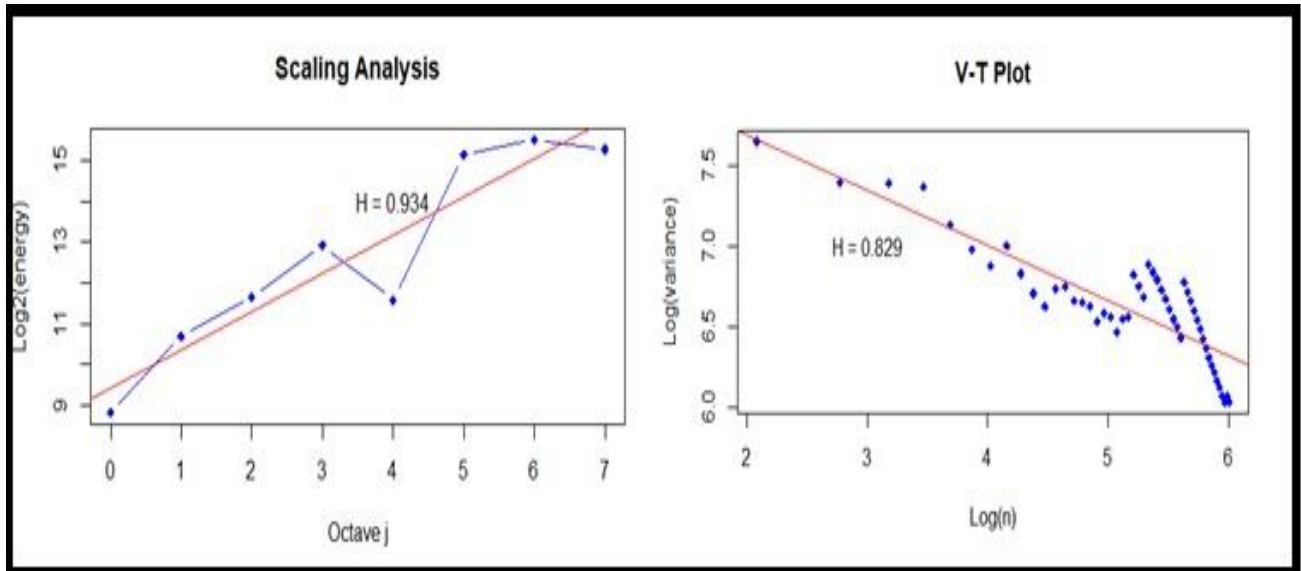


Figure 5.7: Measure of Self-similarity  $h = 0.934, 0.829$  by Wavelet and V-T Method for Facebook packets for 800

Figure 5.7 shows the existence of self-similarity in the dataset, out of 132,099 packets, 800 packets was detected by dpi library are of Facebook. The value of  $h$  is around 0.83 detected by four methods of finding self- similarity.

#### 5.5.1.4 GOOGLE & SELF-SIMILARITY

Google uses secured channel for communication that is https. Google provides browsing, messaging, data archiving and many other services.

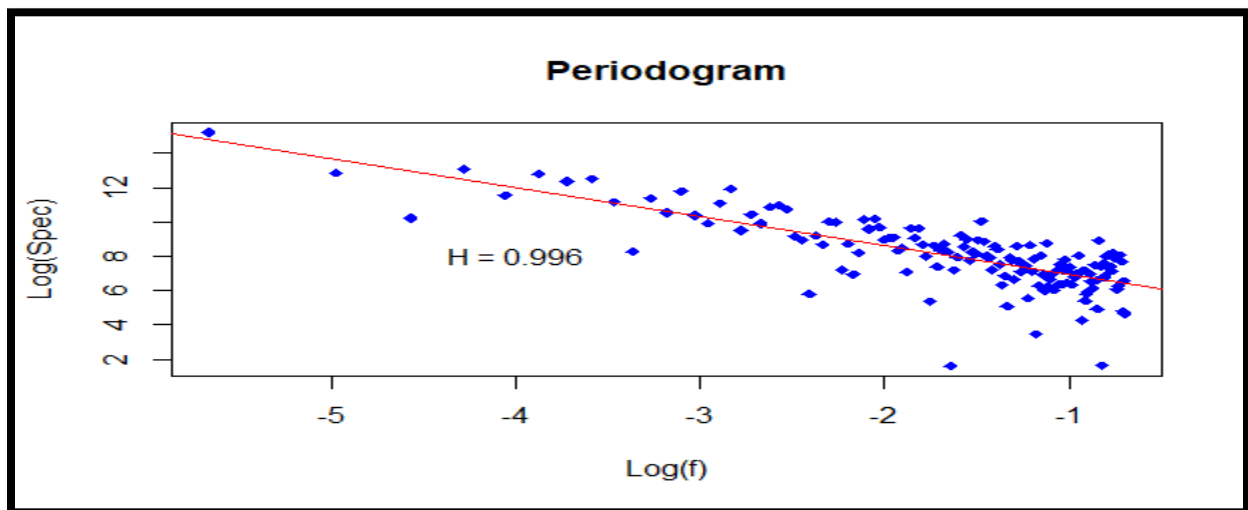


Figure 5.8: Measure of Self-similarity  $h = 0.996$  Periodogram for Google for 800 packets

Figure 5.8 shows the presence of self-similarity in the dataset, out of 132,099 packets, 800 packets was detected by dpi library are of Google. The value of  $h$  is around 0.996 detected by periodogram.

## 5.5.2 SELF-SIMILARITY INDUCED BY INDIVIDUAL PROTOCOLS

### 5.5.2.1 TRAFFIC WITHOUT WHATSAPP & SELF-SIMILARITY

In order to find the impact of separate protocols on over behaviour of tcp flow, aggregated traffic, they are extracted from dataset and self-similarity of remaining dataset is calculated.

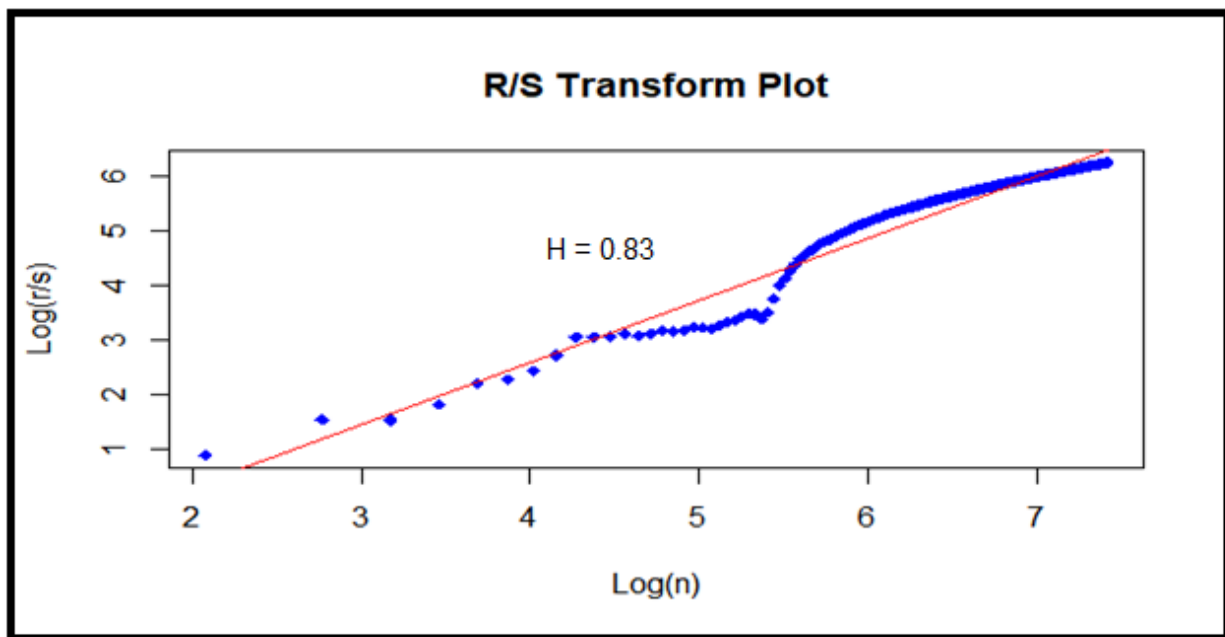


Figure 5.9: Measure of Self-similarity  $h = 0.830$  by R/S Transform without WhatsApp

Figure 5.9 is manifesting a time series of counted packet data and the slope is representing the value of  $H$  which is equal  $h = 0.830$ . As  $h \rightarrow 1$  the data become highly self-similar. However the value of  $h$  of aggregated traffic by R/S Transform 0.989. A variation of 16 % is recorded.

### 5.5.2.2 TRAFFIC WITHOUT WHATSAPP VOICE & SELF-SIMILARITY

How WhatsAppVoice is impacting the aggregated traffic self-similarity is defined in this section. They are selected from dataset and self-similarity of remaining dataset is calculated.

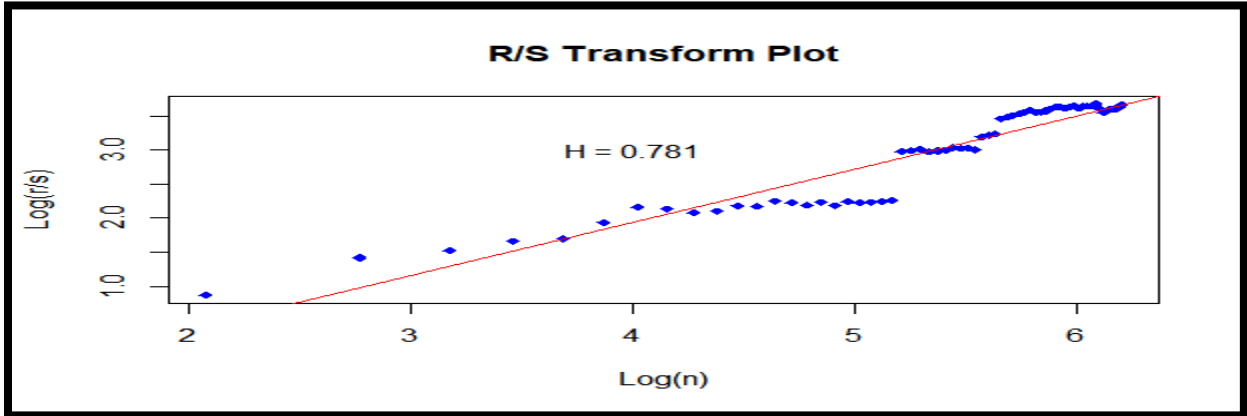


Figure 5.10: Measure of Self-similarity  $h = 0.781$  R/S Transform for without WhatsAppVoice

Figure 5.10 is manifesting a time series of counted packet data and the slope is representing the value of  $H$  which is equal  $h = 0.781$ . As  $h \rightarrow 1$  the data become highly self-similar. However the value of  $h$  of aggregated traffic by R/S Transform 0.989. A variation of 17 % is recorded.

**5.5.2.3 TRAFFIC WITHOUT FACEBOOK & SELF-SIMILARITY**

How Facebook is impacting the aggregated traffic self-similarity is defined in ongoing part. They are selected from dataset and self-similarity of remaining dataset is calculated.

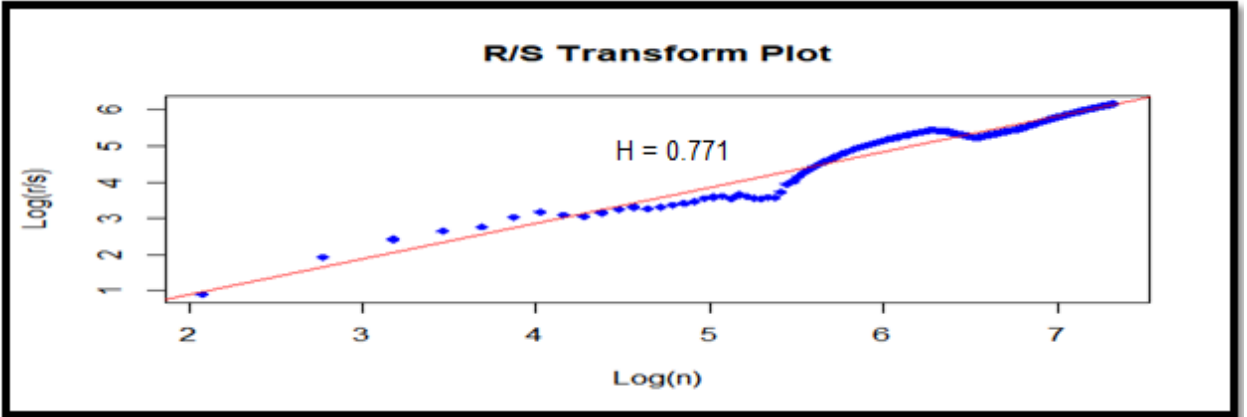


Figure 5.11: Measure of Self-similarity  $h = 0.771$  by R/S transform without Facebook

Figure 5.11 is manifesting a time series of counted packet data and the slope is representing the value of  $H$  which is equal  $h = 0.771$ . As  $h \rightarrow 1$  the data become highly self-similar. However the value of  $h$  of aggregated traffic by R/S Transform 0.989. A variation of 16 % is recorded.



#### 5.5.2.4 TRAFFIC WITHOUT GOOGLE & SELF-SIMILARITY

How Google is impacting the aggregated traffic self-similarity is defined in ongoing part. They are selected from dataset and self-similarity of remaining dataset is calculated.

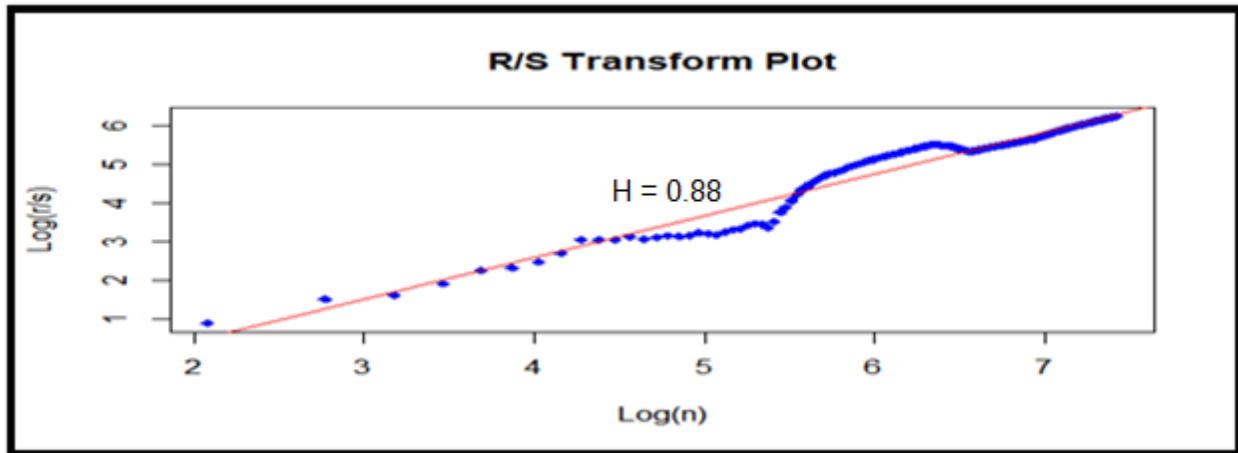


Figure 5.12: Measure of Self-similarity  $h = 0.88$  by R/S transform without Google

Figure 5.12 is manifesting a time series of counted packet data and the slope is representing the value of  $H$  which is equal  $h = 0.88$ . As  $h \rightarrow 1$  the data become highly self-similar. However the value of  $h$  of aggregated traffic by R/S Transform 0.989. A variation of 11 % is recorded.

#### 5.6 RESULTS

Figure 5.13 shows the difference between Hurst parameter when https traffic is part of aggregated traffic and when it is not. The analyzed application layer traffic in this project is WhatsApp, Facebook and Google.

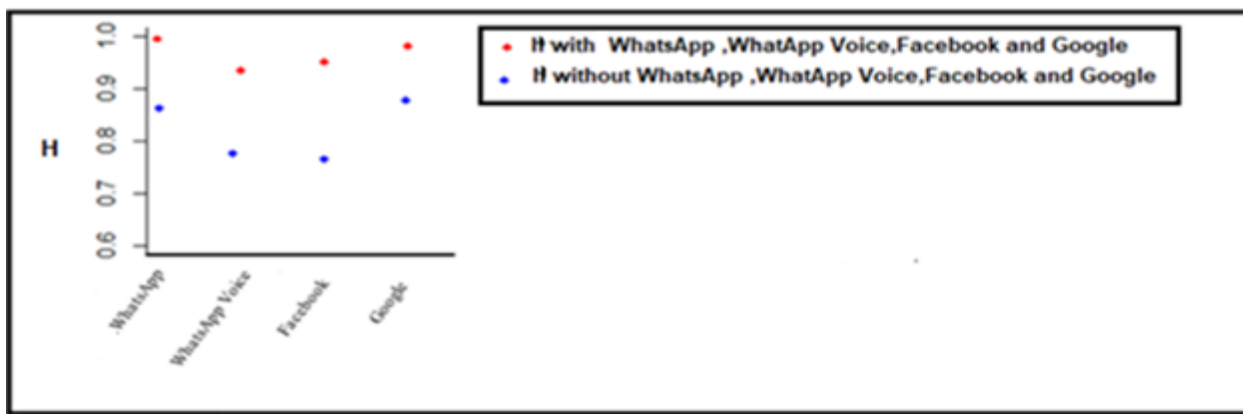


Figure 5.13: Manifesting the Value of Hurst parameter with https Applications and without them; a difference of 11 percent to 17 percent in value of  $H$  is observed.

When packets of such traffic are present they induce more self-similarity. A difference of 15 percent to 21 percent is encountered. As manifested by the figure below. The red dots are value of H when Application layer traffic is part of a trace, they show a value for h around 0.95 as ( $0.5 < h < 1$ ) considered to be a basic condition of self-similarity. The more the parameter tends towards ( $h \rightarrow 1$ ) traffic is considered to be more bursty. On the other hand the blue dots are showing the traffic trace without https Application layer traffic. The trend of H in that is around 0.79, a drop of 18 percent from the previous value of h. This numerical analysis provides a thrust for the claim that application layer trace is making the tcp aggregate more self-similar as it happened to be without them.

### 5.7 INTERNET TRAFFIC ANOMALY DETECTION BASED ON SELF-SIMILARITY

As Self-Similar traffic is a scale invariant and always has the same underlying reference structure. It has been one of the dominant factors of internet traffic modeling. In traffic anomaly such as denial of service (DoS) and distributed denial of services (DDoS) the victims are attacked by flooding a lot of packets in the network at a time. As a result of this victim is unable to get the genuine packets as the network becomes congested. Self-similarity for anomaly detection works on the value of H ( $0.5 < h < 1$ ). If the value is out of range then traffic is called anomalous. One of the tests for self-similar and anomalous data is the Kolmogorov-Smirnov test which gives zero for 0 for normal condition and 1 for anomaly. It is based on the difference of Cumulative distribution and empirical distribution function.

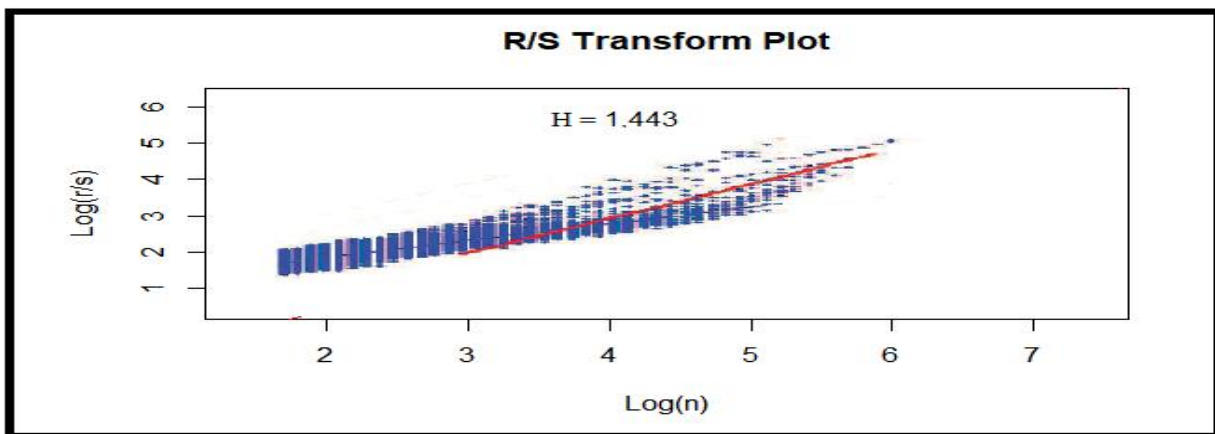


Figure 5.14: Manifesting the Value of Hurst parameter which is 1.443 out of range of ( $0.5 < h < 1$ ) showing the anomalous behaviour of traffic.

Figure 5.14 is depicting another method of finding self-similarity in which self-similarity of data is founded by any one of the twelve methods; in this case it is R/S transform. If value of  $h$  is out of bound of  $(0.5 < h < 1)$  then it is called anomalous data traffic.

## **5.8 SUMMARY OF THE CHAPTER**

Application layer traffic has become one of the dominant factors of inducing self-similarity in the internet traffic. In this chapter analysis of traffic produced by Facebook, WhatsApp and Google is analyzed by one of the methods of finding self-similarity. And a result is concluded which shows 15 to 20 percent more self-similar when traffic from such applications are present. This assertion could help in getting better performance of network and robust use of resources when usage of Facebook and WhatsApp is reduced to get a less bursty traffic. As these two applications induce more self-similarity than Google whose usage is more as it provides one of the widely used search engine, a restriction on the two would allow more room for the protocols of Google to work in a less bursty fashion. Then a section on how to find anomaly from self-similarity is also included to show the application of self-similarity in practical domain. This could be done by finding the value of  $H$ . Which if lies out of bounds of  $(0.5 < h < 1)$  greater than 1 then the traffic trace is called compromised by some DoS or DDoS attacks.

## CHAPTER 6

### CONCLUSION AND FUTURE WORK

To conclude, self-similarity is ubiquitous in the networks and applications that employs. The results in chapter 5 shows that some Applications works on layer seven of OSI model induces more self-similarity as compare to others. They are Facebook and WhatsApp contrary to Google. The value of H for aggregated traffic for the trace files defined in Chapter 1 captured through Wireshark<sup>20</sup> is 0.989 whereas for whatsApp, WhatsApp Voice, Facebook and Google values of H are 0.99, 0.95, 0.93 and 0.99 respectively. When packets of these applications are eliminated from trace one by one and their values of H turn out to be without WhatsApp packets 0.83, without WhatsApp Voice packets, 0.78, without Facebook packets 0.77 and 0.88 without Google packets. A difference of 16 to 17 percent founded when packets of Facebook and WhatsApp are absent and only 11 percent when packets of Google are absent. This assertion could help in getting better performance of network and robust use of resources when usage of Facebook and WhatsApp is reduced to get a less bursty traffic. As these two applications induce more self-similarity than Google whose usage is more as it provides one of the widely used search engine, a restriction of usage on the two would allow more room for the protocols of Google to work in a less bursty fashion. As traffic self-similarity has become one of the dominant factors in the internet traffic modeling. Its application lies in the area of Internet of Things (IoT), Next Generation networks and Future Internet. To study the behaviour of internet of things when billions of devices (IoT) are generating data, a fruit of ubiquitous computing, self-similarity could play an important role in understanding these networks, assessing their performance and proper assignment of resources for them. Peer to peer networks and mobile computing could also employ the procedure of self-similarity measure to ensure better quality of services (QoS) and performance of such networks. Anomaly detection congestion control and resilience of future internet could also be done by using the concepts of self-similarity, burstiness and heavy tail behaviour of them.

---

<sup>20</sup> [www.wireshark.org](http://www.wireshark.org)

## REFERENCES

- [1] A. A. M. Saleh and J. M. Simmons, “Technology and architecture to enable the explosive growth of the internet”, *IEEE Communications Magazine*, 49(1):126–132, January 2011.
- [2] C. Kenjiro, M. Koushirou, and K. Akira, “Traffic data repository at the wide project”, *In Proceedings of the Annual Conference on USENIX Annual Technical Conference*, ATEC '00, pages 51–51, Berkeley, CA, USA, 2000. USENIX Association.
- [3] V. Paxson and S. Floyd, “Wide area traffic: the failure of poisson modeling”, *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.
- [4] B. Mandelbrot, “Self-similar error clusters in communication systems and the concept of conditional stationarity”, *IEEE Transactions on Communication Technology*, 13(1):71–90, March 1965.
- [5] M. E. Crovella and A. Bestavros, “Self-similarity in World Wide Web traffic: evidence and possible causes”, *IEEE/ACM Transactions on Networking*, 5(6):835–846, Dec 1997.
- [6] M. S. Taqqu, W. E Willinger and A Erramilli, “A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks, stochastic networks: Theory and applications”, Oxford University Press, pages 339–420, 1996.
- [7] W. Leland, M. S. Taqqu, W. E. Willinger, and D. V. Wilson, “On the self-similar nature of ethernet traffic (extended version)”, *IEEE/ACM Transactions on Networking*, 2(1):1–15, Feb 1994.
- [8] W. E. Willinger, W. Walter, et al, “Self-similarity in high-speed packet traffic: analysis and modeling of ethernet traffic measurements”, *Statistical Science*, vol. 10, no. 1, 1995, pp. 67-85. JSTOR. [www.jstor.org/stable/2246232](http://www.jstor.org/stable/2246232).
- [9] A. Kolmogorov, “the local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers”, *Proceedings: Mathematical and Physical Sciences*, vol. 434, no. 1890, 1991, pp. 9–13. JSTOR, [www.jstor.org/stable/51980](http://www.jstor.org/stable/51980).
- [10] A. Erramilli and L.W. Jonathan, “Monitoring packet traffic levels”, *IEEE/ACM Transactions on Networking*.
- [11] R. Addie and M. Zukerman, “Fractal traffic: measurements, modeling and performance evaluation”, *Proceedings of IEEE INFOCOM'95*, 1995.
- [12] S. A. Lcock , L Orier , P. & N Elson , R. (2012), “Libtrace: A packet capture and analysis library”, *SIGCOMM Computer Communication Rev.*, 42, 42–48.

- [13] M. S. Taqqu, E. Teverovsky, W. E. Willinger, "Estimators for long range dependence: an empirical study", *Fractals*, 1995, 3, (4), pp. 78.
- [14] J. Beran, "Statistics for long-memory processes" (CRC Press, 1994, 1st edn.)
- [15] P. Abry, D. Veitch, "Wavelet analysis of long-range-dependent traffic", *IEEE Trans. Inf. Theory*, 1998, 44, (1), pp. 2–15.
- [16] N. Scafetta and P. Grigolini, "The thermodynamics of social processes: the teen birth phenomenon", *Fractals*, 2001, 9, (2), pp. 193–208.
- [17] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic", *IEEE Transactions on Communications*, 43(2/3/4):1566–1579, Feb 1995.
- [18] D. E. Duffy, A. A. McIntosh, M. Rosenstein, and W. Willinger, "Statistical analysis of ccsn/ss7 traffic data from working ccs subnetworks", *IEEE Journal on Selected Areas in Communications*, 12(3):544–551, April 1994.
- [19] M. A. Arfeen, K. Pawlikowski, A. Willig, and D. McNickle, "Internet traffic modeling: from superposition to scaling", *IET Networks*, 3(1):30–40, March 2014.
- [20] J. Cao, L. Cleveland and D. Sun, "Internet traffic tends toward Poisson and independent as the load increases", 'Nonlinear estimation and classification' (Springer, New York, 2003), (Lecture Notes in Statistics, 171), pp. 83–109.
- [21] N. G. Duffield and O'Connell, "Large Deviation and overflow probabilities for the general single-server queue, with applications, preprint, dublin institute for advanced studies", *IEEE/ACM Transactions on Networking (TON)* 1993.
- [22] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queuing analysis with long-range dependent packet traffic", *IEEE/ACM Transactions on Networking*, 4(2):209–223, April 1996.
- [23] I. Norros, "A storage model with self-similar input, *Queuing Systems Theory and Applications*", Vol. 16, Issue. 3-4, 1994.
- [24] J. W. Causey and H. S. Kim, "Comparison of admission control schemes in atm networks", *International Journal of Communication Systems*, Vol. 8, 1995.
- [25] M. A. Arfeen, K. Pawlikowski, D. McNickle, and A. Willig, "The role of the weibull distribution in internet traffic modeling", In *Proceedings of the 2013 25th International Teletraffic Congress (ITC)*, pages 1–8, Sep. 2013.
- [26] B. J. West and B. Deering, "The lure of modern science fractal thinking, studies of nonlinear phenomena in life sciences", No. 3, World Scientific Publishing Co., 1995.
- [27] T. Tuan and K. Park, "Congestion Control for Self-Similar Network Traffic", preprint, Purdue University, 1998.

- [28] H-O. Peitgen, H. Jürgens and D.Saupe , “Chaos and Fractals New Frontiers of Science”, springer-Verlag, 1992.
- [29] J. Gleick, “Chaos: Making a new science”, Penguin Books New York, NY, USA, 1987.
- [30] H.E . Hurst, “Long-term storage capacity of reservoirs”, *Trans. Am. Soc. Civil Eng.*, 1951, 116, (3), pp. 770–799.
- [31] J. Beran, “Statistics for Long-Memory Processes”, London: Chapman and Hall, 1994.
- [32] J. W. Magotra and N. T. Stearns, “The compressibility of stationary random processes.”, In 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing” *Conference Proceedings, volume 5*, pages 2527–2530 vol. 5, May 1996.
- [33] S. Sajeed, D. L. Kabir, M. L. Palash, N. Sultana, and S. Rafique, “An approach to measure the hurst parameter for the dhaka university network traffic”, *In 2010 The 7th International Conference on Informatics and Systems (INFOS)*, pages 1–5, March 2010.
- [33] P. Zhangd, “A new fractal point process for modeling self-similar traffic”, In ICCT’98. 1998 *International Conference on Communication Technology. Proceedings (IEEE Cat. No.98EX243)*, volume 2, pages 5 pp. vol.2–, Oct 1998.
- [34] A. Erramilli, J. L. Wang, and W.E. Willinger, “ Self-similar traffic parameter estimation: a semi-parametric periodogram-based algorithm”, *In Proceedings of GLOBECOM ’95*, volume 3, pages 2225–2231 vol.3, Nov 1995.
- [35] G. Samorodnitsky, “Long Range Dependence. Now”, 2007. IEEE/ACM Transactions on Networking.
- [36] M. Grossglauser and J.Bolot, “On the relevance of long-range dependence in network traffic”., *IEEE/ACM Transactions on Networking*, 7(5):629–640, Oct 1999.
- [37] H.E. Hurst, “Long-term storage capacity of reservoirs”, *Transactions of the American Society of Civil Engineers*, 116:770-799, 1951.
- [38] K.Park, & W.E. Willinger, (2002), “Self-Similar Network Traffic: An Overview”, 1–38. John Wiley & Sons, Inc.
- [39] B. B. Mandelbrot and V. V. Ness. Fractional brownian motions, “fractional noises and applications”, *SIAM Review*, 10(4), October 1968.