PROBABILITY DISTRIBUTIONS OF INTER-SAMPLE TIME OF EVENT-BASED SAMPLING ENCODERS

Melanka Sachith Wanniarachchi

168482V

Thesis submitted in partial fulfilment of the requirements for the degree Master of Science in Electronics & Automation

Department of Electronics and Telecommunication Engineering

University of Moratuwa
Sri Lanka

July 2021

DECLARATION

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Masters thesis/Dissertation under my supervision.

Name of the supervisor:

Signature of the supervisor: Date:

ABSTRACT

Probability Distributions of Inter-Sample Time of Event-Based Sampling Encoders

Keywords: Event Based Sampling, Memory Based Event Triggering, Sampling Rate Probability Distribution

Bandwidth is the most important resource in telecommunications. Though recent developments have resulted in a significant increase of available bandwidth, the demand for bandwidth continues to follow and new demands are also created with the introduction of new technologies. Internet of Things is one such development that has resulted in increased demand for bandwidth due to the interconnection of smart sensors and actuators to the Internet.

Increased demand for limited bandwidth results in congestion which can in tern negatively affect the reliability of the network by causing latency (delay), jitter (delay variation) and data loss (in the form of packet drops). Event based sampling is a strategy of mitigating congestion that does so by reducing network traffic. This is achieved by reducing the effective sampling rate and it is highly successful if the signal exhibits high dependency between samples. Despite numerous empirical studies, no attempt has been made to obtain a probability distribution of the traffic rate of such encoders. This study aims to obtain such a model for a type of event-based sampling known as memory-based event triggering.

With a statistical model of the generated traffic, it is possible to get an idea about the network capabilities and effectively mitigate the congestion. Correctness of the statistical model can be verified by the empirical results and it is possible to easily determine the maximum number of sensors for a given network bandwidth with a given quality of service.

DEDICATION

To My Parents, Wife, Teachers and who ever helped.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor Dr. Upeka Premaratne, Senior Lecturer at Department of Electronic and Telecommunication Engineering, University of Moratuwa for his help given to me in the form of ideas and concepts, alternative methods, feedback, guidance and motivation.

Secondly, I would like to thank Dr. Nuwan Dayananda, Head of the Department of Electronic and Telecommunication Engineering and Dr. Chamira Edussooriya, Course Co-ordinator, PG Dip / MSc in Electronics and Automation, Dept. of electronic and Telecom Engineering for their valuable support and guidance.

Furthermore, I would like to extend my gratitude to Mr. Damith Kandage, Course Assistant, PG Dip / MSc in Electronics and Automation for his support.

Additionally, I would like to thank Technical Officers Mr. Thisara Wickramasinghe, Mr. Chintaka Ranawaka, Mr. Sameera Fernando and Mr. Weditha Dissanayake along with Lab Attendants Mr. Chaminda Kaluarachchi, Mr. Sumudu Perera and Mr. Peter Ferdiando of the Department of Electronic and Telecommunication Engineering for facilitating the experiments.

Finally, I would like to thank the Managing Director of Synergen Technology Labs PVT Ltd. For his support to make this project a success.

M. S Wanniarachchi
Department of Electronic and Telecommunication Engineering,
University of Moratuwa,
Sri Lanka.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADM	Adaptive Delta Modulation
DEM	Deadband Error Modulation
ETADM	Event Triggered Adaptive Differential Modulation
MBET	Memory Based Event Triggering
MET	Memory-less Event Triggering
PBET	Prediction-Based Event Triggering
KLD	Kullbeck-Liebler Divergance
TVD	Total Variation Distance
IoT	Internet of Things
IIoT	Industrial Internet of Things

CHAPTER 1

1. INTRODUCTION

1.1 Overview

Bandwidth availability and reliability of telecommunication networks are essential for the proper accessibility of cloud computing [1]. Numerous recent developments in the terahertz band [2] [3] [12] [44] have seen a sharp rise in available bandwidth for wired and wireless communications. However, similar to Parkinson's law on work expanding to fill up the existing deadline, demand in telecommunication networks grows to fill the available bandwidth [13]. An example for this is Wireless Multimedia Sensor Networks (WMSN) [14] [15] for sensing and surveillance and the resulting congestion due to bandwidth constraints [16] [17] [18]. Other emerging high demand applications include Internet of Things [19] [20] smart cities [43] and Industry 4.0 [21].

The promising potential of the emerging Industrial Internet of Things (IIoT) and Internet of Things (IoT) technologies are playing an important role in the modern network architectures. The number of Internet of Things devices worldwide is forecast to almost triple from 8.74 billion in 2020 to more than 25.4 billion IoT devices in 2030.

Congestion can result in higher latency [22] [23] and data loss due to packet drops [24] [25] [26]. Latency is critical in real time, safety critical and quality of service-oriented cloud applications [27] [28]. The primary motivation for edge computing (Edge computing is a distributed computing paradigm that brings computation and data storage closer to the location where it is needed, to improve response times and save bandwidth.) is latency reduction [29] [30]. Thus, congestion mitigation must go hand in hand with it for its full benefit to be realized. Furthermore, in fog computing applications with unreliable communication links, the limited capability of the link will be overwhelmed by congestion.

Compared with the widely adopted time-driven schemes, these event-based samplers have advantages of improving the efficiency in resource utilization in many resource constrained applications. Event based sampling techniques that reduce the effective sampling rate by encoding the high dependence of sensory, control and actuator signals are viable solutions for congestion mitigation. In the method proposed in [6] sampling at the input occurs periodically at the Nyquist or sampling rate that is required to maintain control system stability [31]. However, a transmission of data over the network takes place only when there is a significant difference in the input that exceeds a given threshold. This results in an effective sampling rate that is less than the

periodical sampling rate making it highly desirable for cloud computing nodes [32]. To illustrate, in a datacentre, an ambient temperature change of 0.1^{o}C may not be significant but a change of 1 ^{o}C or 5 ^{o}C may indicate the formation of a potential hotspot or equipment thermal runaway and require data transmission while no or small changes such as that of 0.1 ^{o}C can be ignored reducing the traffic burden. In methods such as [33] [34] [35], the bandwidth is significantly reduced by encoding the difference between two successive samples. However, the periodic sampling rate remains unchanged. When compared to congestion avoidance methods that involve caching [36] [37], event-based sampling methods such as [6] are more resource efficient in terms of algorithmic implementation and not needing to negotiate a protocol or cache data. The trade-off for this benefit is at the expense of reduced accuracy [44].

1.2 Problem Statement

Currently, the most efficient methods of encoding are that of [6] (which was named as Memory Based Event Triggering (MBET) in [7]) and Deadband Error Modulation (DEM) [8]. In [9] based upon the mutual information of the input and output it was demonstrated that MBET is superior to DEM for pure sampling applications. This makes it a viable solution for congestion mitigation in delay critical edge computing. However, in order to effectively mitigate congestion a statistical model of the generated traffic is required. To the best of the authors' knowledge no previous attempt has been made to determine such a statistical model which will allow useful quantifications to be made. An example model is shown in Figure 1.1. For example, such model would enable the maximum number of sensors for a given network bandwidth and required quality of service to be determined. Such a question would be vital in safety critical cloud-based applications such as supervisory control of petroleum refineries [10].

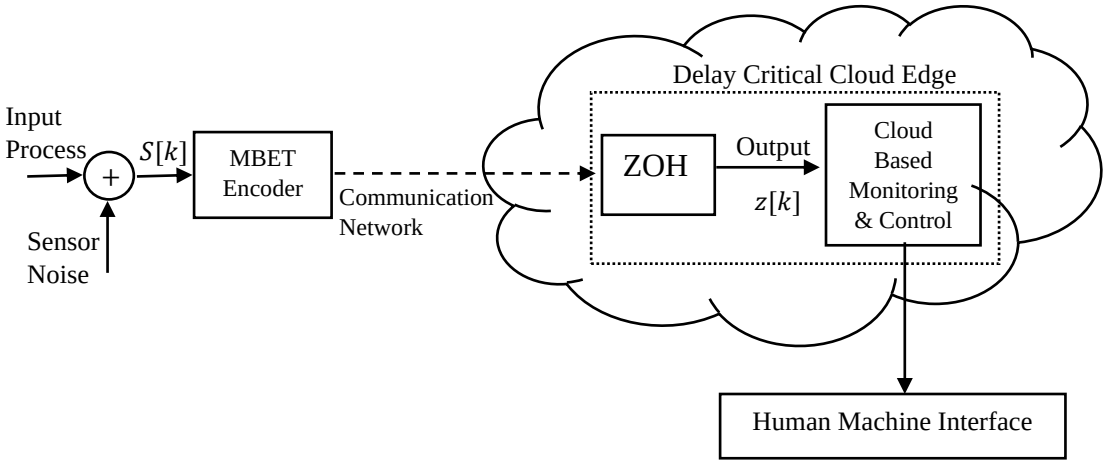

Figure 1.1: MBET Encoder Implementation in Delay Critical Cloud Edge

1.3 Outline of Thesis

In brief this thesis discusses about a statistical modal of the generated traffic of an efficient event-based sampling encoder. In Chapter 1, the importance of the bandwidth and the cruciality of reducing the packets entering to the network in order to efficiently use the available bandwidth are discussed. In Chapter 2, the available event-based sampling encoders are described. In the next chapter, a model is developed to get the theoretical probability distribution of the traffic rate of a memory-based event triggering encoder. Chapter 4 is used to present the empirical results and discussion. In Chapter 5, future directions are discussed.

CHAPTER 2

2. LITERATURE SURVEY

Since high bandwidth applications such as online surveillance and IoT are gaining popularity there is a rapid requirement for bandwidth. In the near future, one of the prime factors that need to be addressed is the increased bandwidth [2][3]. However, reducing the number of packets entering to the network seems to be a viable solution for this [4]. These event-based sampling encoders execute only when a specific event is occurred (i.e., Sampling a variable when a threshold is exceeded instead of periodic sampling).

Prominent event-based encoders are

- Memory-less Event Triggering (MET) [7]
- Memory Based Event Triggering [11]
- Event Triggered Adaptive Differential Modulation (ETADM) [7]
- Deadband Error Modulation [8]
- Prediction Based Event Triggering [45]

2.1 Memory-less Event Triggering

In the memory- less event triggering encodes, signal is transmitted only when the input variable exceeds a predetermined threshold [5]. Triggering mechanism and the schematic are illustrated in Algorithm 1 and Figure 2.1 respectively. *v[k]* is the input variable to the encoder at time *k*.

IF | *v[k]* | > *Threshold*

Then

Transmit v[k]

END

Algorithm 1: Triggering mechanism for MET

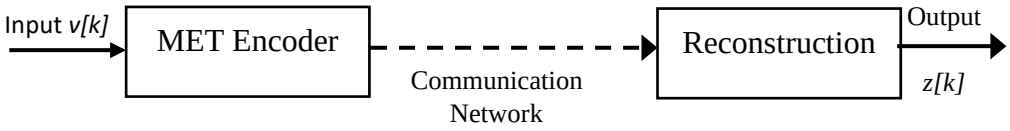

Figure 2.1: Schematic of MET

It is important to mention that MET is highly inefficient because it is transmitting data only if the input signal exceeds the threshold value. In all scenarios it is transmitting at the Nyquist sampling rate. Therefore, it is only suitable for rare signals that are mostly zero.

2.2 Memory Based Event Triggering

In memory-based event triggering encoders [6], a memory is used to store the value of the input at the last sampling event. In Addition to the memory-based event triggers, a zero-order hold is needed at the receiver side to reconstruct the signal. Triggering mechanism and the schematic for MBET are illustrated in Algorithm 2 and Figure 2.2 respectively. Here $x[k]$ is the variable input value at time k and x_M is the memory value. e_T is the event triggering threshold. $z[k]$ is the reconstructed output and ideally $z[k]$ should be same as $x[k]$.

IF $|\, x[k] - x_M \,| > e_T$ **Then**
    $x_M = x[k]$ // store current input
    Transmit (x_M)
else
    //Do nothing
END

Algorithm 2: Triggering Event for MBET Encoder

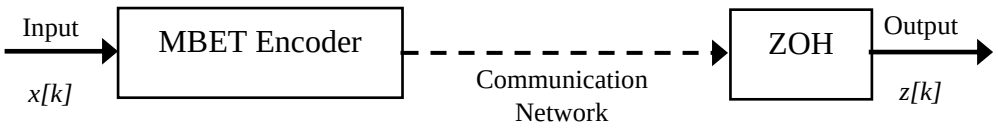

Figure 2.2: Schematic of MBET

2.3 Event Triggered Adaptive Differential Modulation

The paper [7] proposes Event Triggered Adaptive Differential Modulation. ETADM encoder uses Adaptive Delta Modulation (ADM).

The ADM encoder is used to encode the difference into a variable step size $s[k]$. The step size is used as the input for event triggering. If the step size exceeds the event triggering threshold, a sampling event is triggered and it is transmitted over the

communication network. Here *v[k]* is the variable input to the encoder at time *k. s[k]* is the ADM encoder output and *z[k]* is the ETADM output.

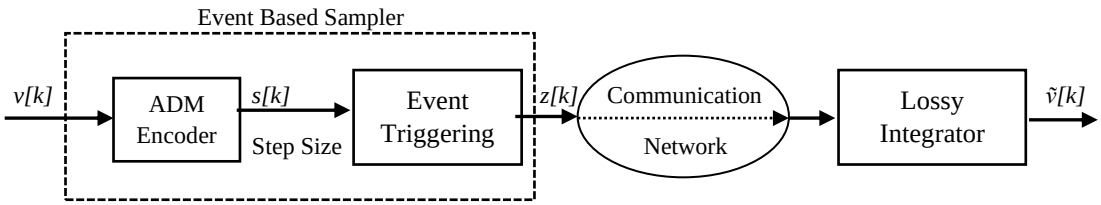

Figure 2.3: ETADM Signal Reconstruction of a Sample Sensor Output

At the receiver end of the channel, the signal is reconstructed by a lossy integrator instead of a ZOH. The lossy integrator makes ETADM robust to bounded packet drops compared to MBET [38]. Figure 2.3 illustrate the signal reconstruction of an ETADM encoder.

2.4 Deadband Error Modulation

Deadband Error Modulation [8] is a tri-level encoder. It's a simpler implementation compared to ETADM. DEM consists of two lossy integrators in the encoder and decoder. The triggering mechanism and the schematic of DEM is shown in Algorithm 3 and Figure 2.4 respectively.

Here *x[k]* is the variable input to the encoder at time *k*, *s[k]* is the output of the encoder, *z[k]* is the lossy integrator output at time *k*, *z[k]* is the reconstructed signal and *d* is the threshold value.

IF $|x[k] - z[k]| > d$ **then**
    $s[k] = s$ // and transmit
else if $z[k] - x[k] > d$ **then**
    $s[k] = -s$ //and transmit
else
    //Do nothing
END
Calculate $z[k+1] = Kz[k] + Ks[k]$ // the output of the lossy integrator

Algorithm 3: Triggering mechanism of DEM

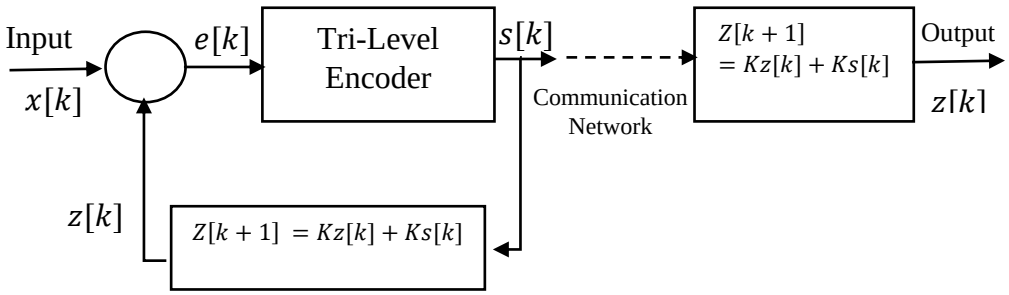

Figure 2.4: Schematic of DEM, Source: [9]

Guided by the results from [8] it is obvious that MBET and DEM are the most efficient encoding methods. Then from the [9] it shows that MBET is superior to DEM for pure sampling.

CHAPTER 3

3. METHODOLOGY

3.1 Sensor Input and Random Process

The sensor input random process (input Y_k in Figure 3.1) is modelled as a Gaussian random walk with Gaussian sensor noise

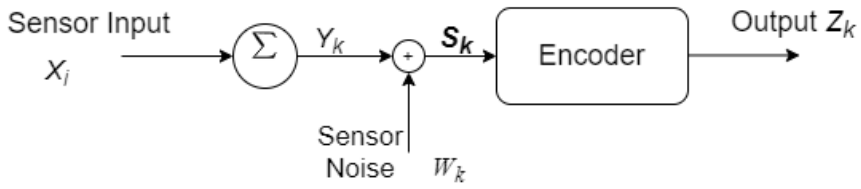

Figure 3.1: Random Process Input Model

$$Y_k = \sum_{i=0}^{k} X_i$$

$$S_k = \sum_{i=0}^{k} X_i + W_k \qquad (1)$$

where sensor input (X_i) are i.i.d. steps of the process with X_i being a normal distribution with mean μ_P and variance σ_P ($X_i \sim N(\mu_P, \sigma_P)$) and sensor noise ($W_k$) is also a normal distribution with zero mean and variance σ_P ($W_k \sim N(0, \sigma_S)$). Input to the encoder is

$$S_k = Y_k + W_k$$

and accounts for the sensor noise such that S_k follows a linear trend (corresponding to the measured parameter) with

$$E[S_k] = k\mu_p + s[0] \qquad (2)$$

$s[0]$ is the initial value. Typically, initial value is taken as zero.

3.2 Sample Model

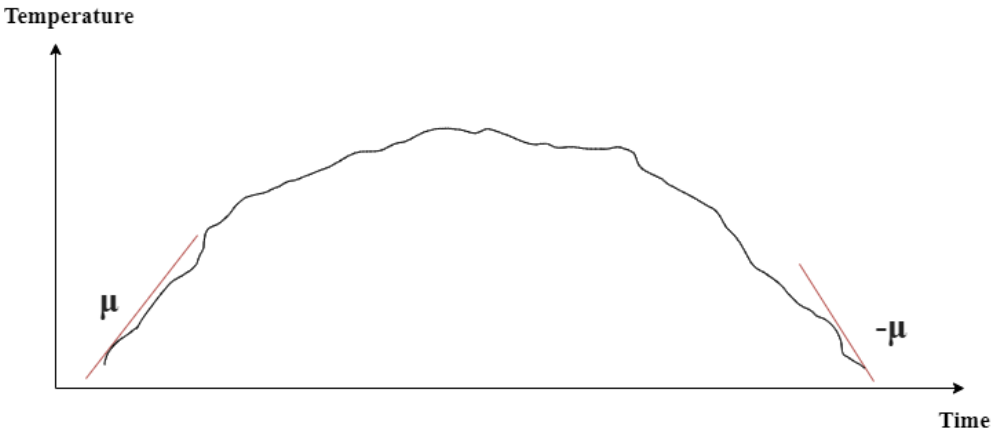

Figure 3.2: Sensor Model

$$\mu_p \in (-\mu, \mu)$$

Sensor mean value μ is considered to be a small number such that $e_T >> \mu$. For an example a typical temperature sensor value graph is shown in Figure 3.2. For a very small segment of the graph the mean value can be considered as a constant. The model focuses on the two extremes of μ_p, i.e, $-\mu$ and μ which will typically represent the largest rate of change in the sensor and in return the highest traffic rate. This is achieved by considering the traffic distribution as a Folded Normal Distribution.

For an event to be triggered,

$$|S_k| \geq e_T$$

In order for the result to be realistic, the process and sensor parameters (μ_P , σ_P and σ_S) can be varied to make the system heterogeneous.

3.3 Inter Event Distribution

Let m be the time which the last event was triggered. Let the time from that point onward be $k+m$. Therefore, the time k is reset after the last event m is triggered. So, k will start from zero when $m = 0$. Let the triggering of an event at sampling time k and $m+k$ is denoted by S_k and S_{m+k}. We can analyze the distribution between two events by investigating the Y_k and Y_{m+k}.

$$Y_{m+k} - Y_m = \sum_{i=0}^{m+k} X_i \; - \sum_{i=0}^{m} X_i$$

$$Y_{m+k} - Y_m = \sum_{i=0}^{k} X_i \qquad (3)$$

When the noise from the sensor is added,

$$S_k = Y_k + W_k \qquad (4)$$

Therefore, let the probability of an event being triggered after *k* time samples is given by π_k and,

When *k* = 1,

$$\pi_1 = P(|S_1| \geq e_T)$$

When *k* = 2,

$$\pi_2 = P(|S_1| < e_T) * P(|S_2| \geq e_T)$$

When *k* = 3,

$$\pi_3 = (1 - \pi_1) * (1 - \pi_2) * P(|S_3| \geq e_T)$$

When *k* = *k*, Therefore, for an event to be triggered after arbitrary time *k* samples

$$\pi_k = P(|S_k| \geq e_T) \prod_{k=1}^{k-1}(1 - \pi_k)$$

$$\pi_k = P\left(\left|\sum_{i=0}^{k} X_i + W_s\right| \geq e_T\right) \prod_{k=1}^{k-1}(1 - \pi_k) \qquad (5)$$

Given a normally distributed random variable *X* with mean μ and variance σ^2, the random variable $Y = |X|$ has a folded normal distribution. Then the probability density function (PDF) given by,

$$f_Y(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)}{2\sigma^2}} + \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x+\mu)}{2\sigma^2}}$$

For a normally distributed random variable *X* with mean μ and variance σ^2, Cumulative Distribution Function (CDF) is given by

$$F_X(x;\mu,\sigma^2) = P(X \le x)$$

We can calculate the CDF for the folded normal distribution Y as,

$$F_Y(x;\mu,\sigma^2) = P(Y \le x) = F_X(x;\mu,\sigma^2) + F_X(x;-\mu,\sigma^2)$$

In statistics, Q-function gives the tail probabilities of a normal distribution.

$$Q(x) = P(X>x) = 1 - CDF(X) \qquad (6)$$

Hence,

$$F_Y(x;\mu,\sigma^2) = P(Y \le x) = 1 - \left(Q\left(\frac{x-\mu}{\sigma}\right) + Q\left(\frac{x+\mu}{\sigma}\right)\right)$$

$$P(Y > x) = 1 - P(Y \le x)$$

$$P(|X| > x) = 1 - \left[1 - \left(Q\left(\frac{x-\mu}{\sigma}\right) + Q\left(\frac{x+\mu}{\sigma}\right)\right)\right]$$

$$= Q\left(\frac{x-\mu}{\sigma}\right) + Q\left(\frac{x+\mu}{\sigma}\right)$$

Applying this to $P\left(\left|\sum_{k=0}^{k} X_k + W_s\right| \ge e_T\right)$,

$$P\left(\left|\sum_{i=0}^{k} X_i + W_s\right| > e_T\right) = Q\left(\frac{e_T - k\mu_p}{\sqrt{k{\sigma_p}^2 + {\sigma_s}^2}}\right) + Q\left(\frac{e_T + k\mu_p}{\sqrt{k{\sigma_p}^2 + {\sigma_s}^2}}\right) \qquad (7)$$

Since the Q-function is used in the computation of π_k, there is a high potential error when e_T is large and falls within the tail of the distribution. Therefore, it is necessary to find an accurate range for π_k. Therefore, from equation (5) of [40] and expansion for small x,

$$erfc(x) \le e^{-x^2}$$

$$Q(x) \le \frac{1}{2} e^{-\frac{1}{2}x^2} \approx \frac{1}{2}\left[1 - \frac{1}{2}x^2\right] \qquad (8)$$

Therefore,

$$P\left(\left|\sum_{i=0}^{k} X_i + W_s\right| > e_T\right) = 1 - \frac{1}{2}\frac{e_T^2 + \left(k\mu_p\right)^2}{\left(k\sigma_p^2 + \sigma_s^2\right)}. \qquad (9)$$

From (9) for any $j \in \mathbb{Z}^+$ such that $0 < j < k$,

$$1 - P\left(\left|\sum_{i=0}^{j} X_i + W_s\right| > e_T\right) = \frac{1}{2}\frac{e_T^2 + (j\mu_p)^2}{(j\sigma_p^2 + \sigma_s^2)} < 1 \qquad (10)$$

which results in the supremum of e_T occurring for $j = 1$ and given by,

$$e_T < \sqrt{2(\sigma_P^2 + \sigma_S^2) - (\mu_p)^2} \qquad (11)$$

Therefore, for the resulting distribution to be accurate e_T has to be less than the dominant resultant variance of the input process and sensor noise. Since the MBET encoder can be approximated by a biased quantizer according to [41], this results in an increased accuracy of the reconstruction at the expense of lesser traffic reduction.

Then we can construct the probability mass function (pmf) of the inter-event distribution which is denoted by Λ. π_k is a product of numbers less than unity that is asymptotically zero, it has to be truncated at k_{max} for small arbitrary threshold ϵ such that $\pi_k \leq \epsilon$ for all $k \leq k_{max}$. Typically, ϵ can be a very small number. For fixed process and sensor parameters the tradeoff curve between expected value of Λ and average traffic rate $\lambda(e_T)$ can then be obtained as follows.

$$\lambda(e_T) = 1 / E[\Lambda(e_T)]$$

From the weak law of large numbers for a given e_T,

$$\lambda(e_T) = \frac{1}{\mathbb{E}[\Lambda(e_T)]} \approx \frac{\eta_e}{\eta_p} = p(V = 1) \qquad (12)$$

where η_e is the number of packets generated by the MBET encoder, η_p, the number of periodic samples and V are Bernoulli traffic source. Here we have calculated the average traffic in two ways, using the inter-event distribution as well as using the number of packets generated. Previous works such as [42] have heuristically approximated event-based sampling to a Bernoulli traffic source without the theoretical justification.

3.4 Congestion Calculations

Once λ of the encoder is known from (12), it is now possible to determine the congestion due to a number of sensors n for a given bandwidth B. From (12) it is now possible to model the combined traffic of all encoders as a sum of Bernoulli traffic sources where

$$H_n = \sum_{i=0}^{n} V_i \qquad (13)$$

For a given congestion threshold tB where 0 < t < 1 and commonly t = 0.75, the probability of exceeding this threshold, $p(H_n > tB)$ will be of interest.

3.4.1 Homogeneous Sensors

In the homogeneous case all of the input process, sensor and encoder parameters are identical. The sum of (13) becomes Binomial for small n and Gaussian at the asymptote

according to the De Moivre-Laplace theorem where $H_n \sim N(n\lambda;\ n\lambda(1-\lambda))$. Therefore, from (12) and (8) where only the bound is considered,

$$p(H_n > tB) = Q\left(\frac{tB-n\lambda}{n\lambda(1-\lambda)}\right) \leq \frac{1}{2}\, e^{-\frac{1}{2}\left(\frac{tB-n\lambda}{\sqrt{n\lambda(1-\lambda)}}\right)^2} \qquad (14)$$

Therefore, for a given quality-of-service probability for a given threshold, $p_D = p(H_n > tB)$ (14) can be solved to obtain the maximum number of sensors n_{max}.

n_{max} can be obtained by taking the natural logarithm both sides for equation 14,

$$(n\lambda)^2 + 2n\lambda[(1-\lambda)\, ln\, 2p(H_n > tB) - tB] + (tB)^2 \leq 0$$

$$\text{So, } n_{max} = \frac{2\lambda[(1-\lambda)\, ln\, 2p(H_n>tB)-tB] + \sqrt{(2n\lambda[(1-\lambda)\, ln\, 2p(H_n>tB)-tB])^2-4\lambda^2(tB)^2}}{2\lambda^2} \qquad (15)$$

Here when getting the roots, only the addition will be considered because the root with the negation will be small.

CHAPTER 4

4. RESULTS

In this section the probability model obtained in Section 3 is empirically validated by simulation using OMNeT++ Discrete Event Simulator. OMNET++ is a simulation library and framework used for building network simulators. The empirical probability mass function generated by a single encoder is compared with the theoretical values using the Kullbeck-Liebler Divergance (KLD) and Total Variation Distance (TVD) for a selected set of values of σ_P, σ_S, μ_P and e_T.

4.1 Single Sensor Simulations

Table 4.1: Probability Distribution Validation for Constant μ_P (μ_P = 0.01)

σ_P	σ_S	(11) RHS	Metric	e_T				
				1	2	4	8	16
0.5	0.5	1.00	TVD	0.21	0.52	0.90	1.23	1.47
			KLD	0.17	0.94	4.37	16.67	inf
1	0.5	1.58	TVD	0.16	0.37	0.73	1.10	1.41
			KLD	0.09	0.56	2.65	11.93	53.10
2	0.5	2.92	TVD	0.07	0.20	0.46	0.86	1.17
			KLD	0.02	0.13	0.82	3.89	15.35
3	1	4.47	TVD	0.03	0.09	0.27	0.57	0.96
			KLD	0.01	0.04	0.26	1.40	6.34

First set of simulation was done for a constant μ_P and several e_T values. σ_P and σ_S values were increased gradually to see the behaviour of KLD and TVD values. The obtained results are presented in the Table 4.1.

Table 4.2: Probability Distribution Validation for Variable μ_P

(σ_P,σ_S)	μ_P	(11) RHS	Metric	e_T				
				1	2	4	8	16
(1,0.5)	0.05	1.58	TVD	0.15	0.38	0.71	1.00	1.20
			KLD	0.08	0.55	2.50	9.20	24.05
	0.1	1.58	TVD	0.16	0.36	0.63	0.89	0.96
			KLD	0.08	0.50	2.01	5.70	7.40
	0.2	1.57	TVD	0.16	0.32	0.51	0.69	0.88
			KLD	0.06	0.34	1.04	1.70	2.08
(2,0.5)	0.05	2.92	TVD	0.07	0.19	0.46	0.82	1.15
			KLD	0.02	0.13	0.78	1.30	13.99
	0.1	2.91	TVD	0.07	0.19	0.45	0.78	1.06
			KLD	0.02	0.13	0.80	3.61	11.05
	0.2	2.91	TVD	0.07	0.18	0.42	0.69	0.88
			KLD	0.02	0.12	0.68	2.66	5.50

Then the KLD and TVD values were obtained by changing the μ_P value also. The results for this simulation is presented in the Table 4.2.

4.1.1 Examination of the Behaviour of λ_{Prac} and λ_{Theo} Values.

While keeping the σ_P, σ_S, μ_P values fixed, several simulations were done while changing the e_T value to examine the behaviour of the $\lambda_{Practical}$ and $\lambda_{Theoritical}$ values. Here y axis is the probability values and x axis is the k value.

Simulation for $\sigma_P = 1$, $\sigma_S = 0.5$ and $\mu_P = 0.01$

From equation 11 we have shown that in order to resulting distribution to be accurate, e_T has to be a small number. According to the equation 11, for values of $\sigma_P = 1$, $\sigma_S = 0.5$ and $\mu_P = 0.01$, e_T should be less than 1.5811(from equation 11). Figure 4.1, Figure 4.2, Figure 4.3, Figure 4.4 and Figure 4.5 show the Probability Mass Function for different e_T values and deviation of the two models when the e_T value increase is clearly visible.

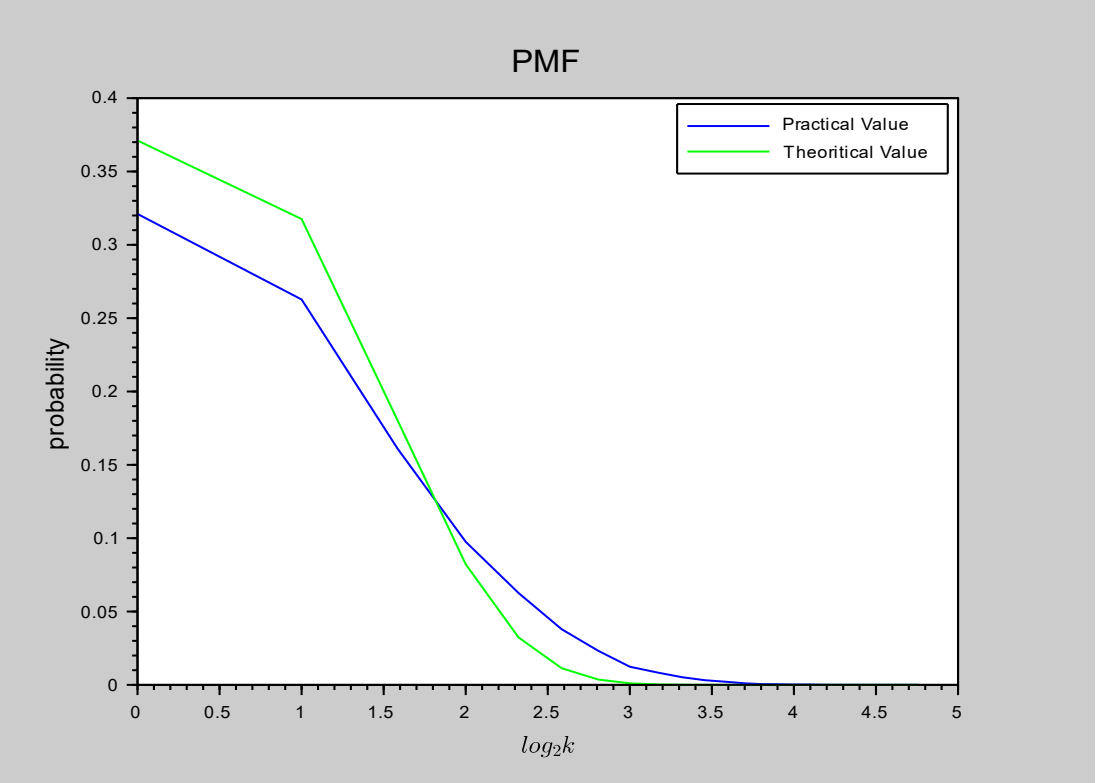

Figure 4.1: Comparison of Distributions ($e_T = 1$, $\sigma_P = 1$, $\sigma_S = 0.5$ and $\mu_P = 0.01$)

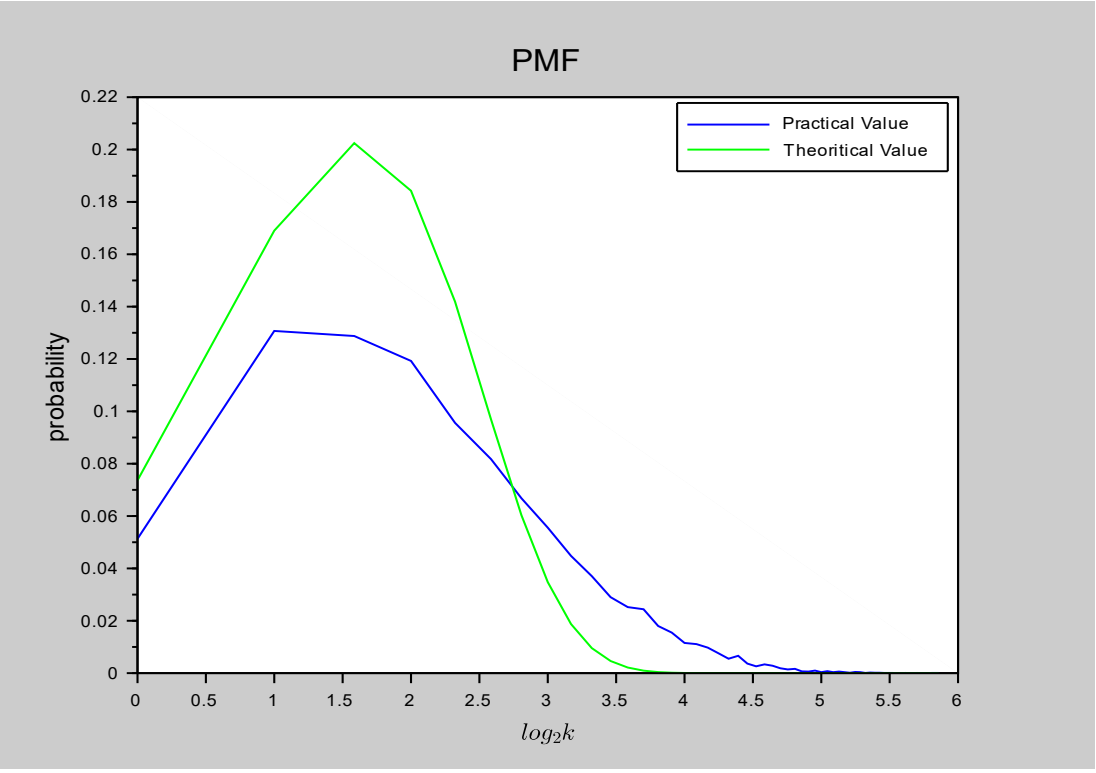

Figure 4.2: Comparison of Distributions ($e_T = 2$, $\sigma_P = 1$, $\sigma_S = 0.5$ and $\mu_P = 0.01$)

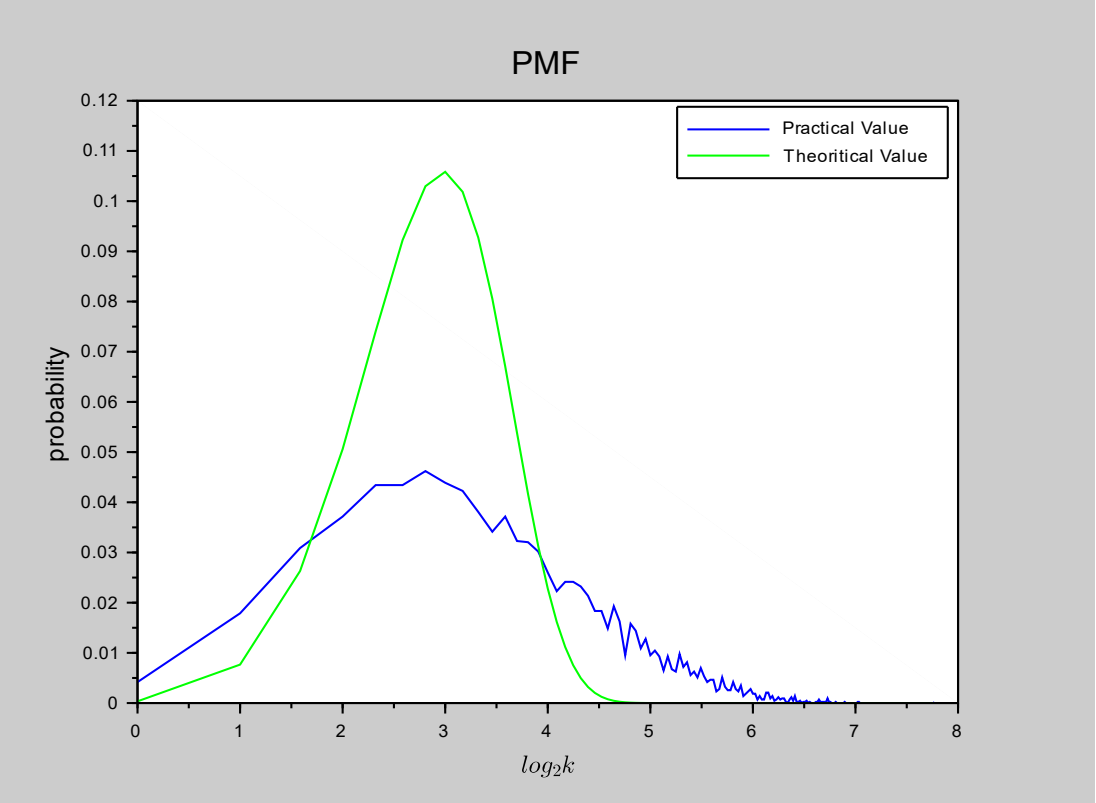

Figure 4.3: Comparison of Distributions ($e_T = 4$, $\sigma_P = 1$, $\sigma_S = 0.5$ and $\mu_P = 0.01$)

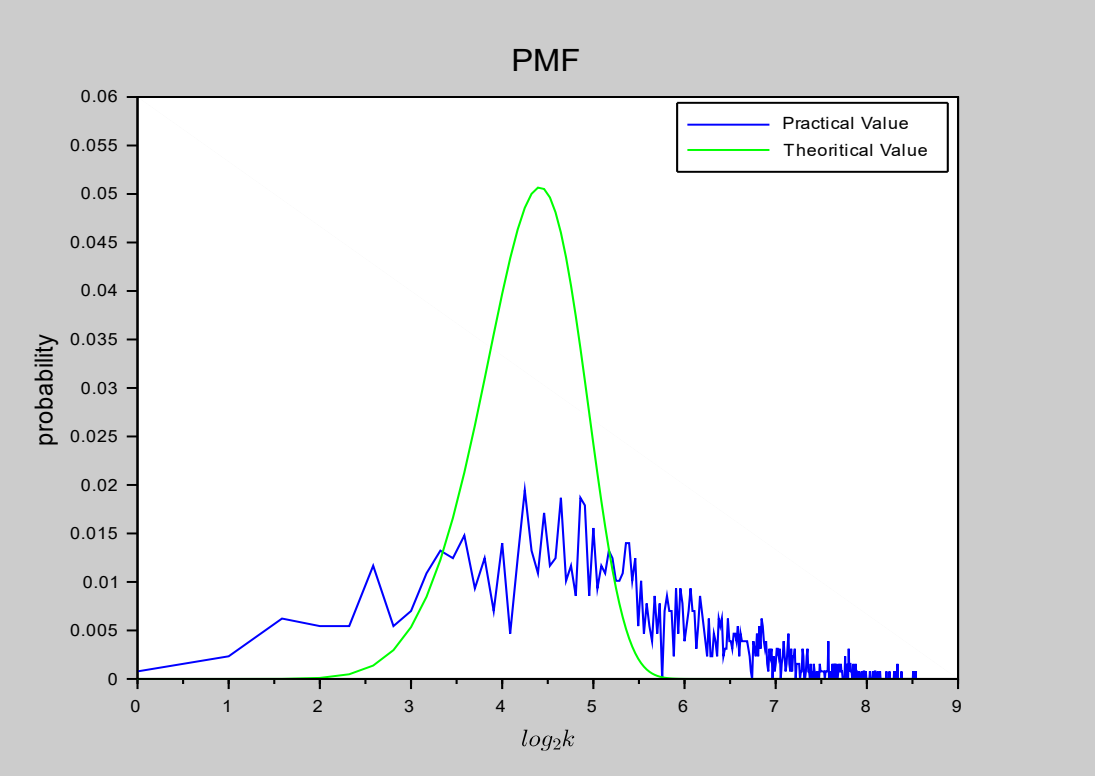

Figure 4.4: Comparison of Distributions ($e_T = 8$, $\sigma_P = 1$, $\sigma_S = 0.5$ and $\mu_P = 0.01$)

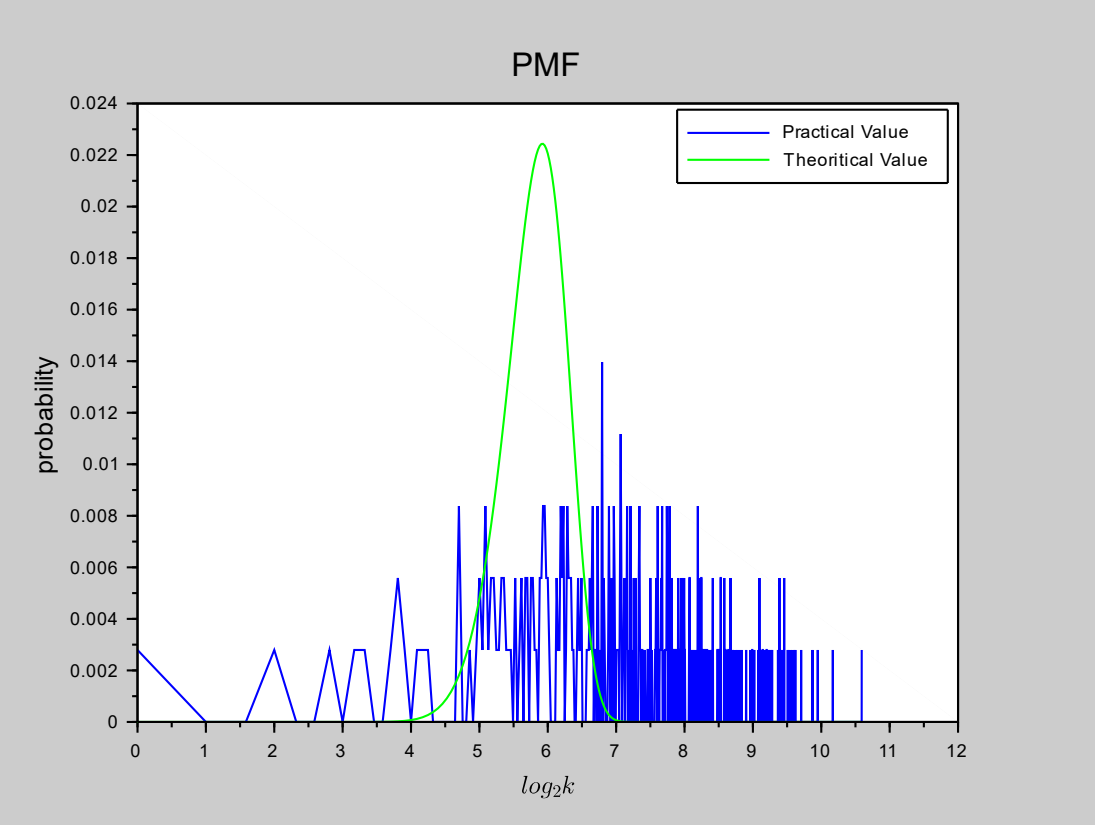

Figure 4.5: Comparison of Distributions (e_T = 16, σ_P = 1, σ_S = 0.5 and μ_P = 0.01)

From the above graphs we can clearly see that when the e_T value is increased, two distributions tend to differ.

Simulation for σ_P = 2, σ_S = 1 and μ_P = 0.01

According to the equation 11, for values of σ_P = 2, σ_S = 1 and μ_P = 0.01, e_T should be less than 3.1623. Figure 4.6, Figure 4.7, Figure 4.8, Figure 4.9 and Figure 4.10 show the Probability Mass Function (PMF) for different e_T values.

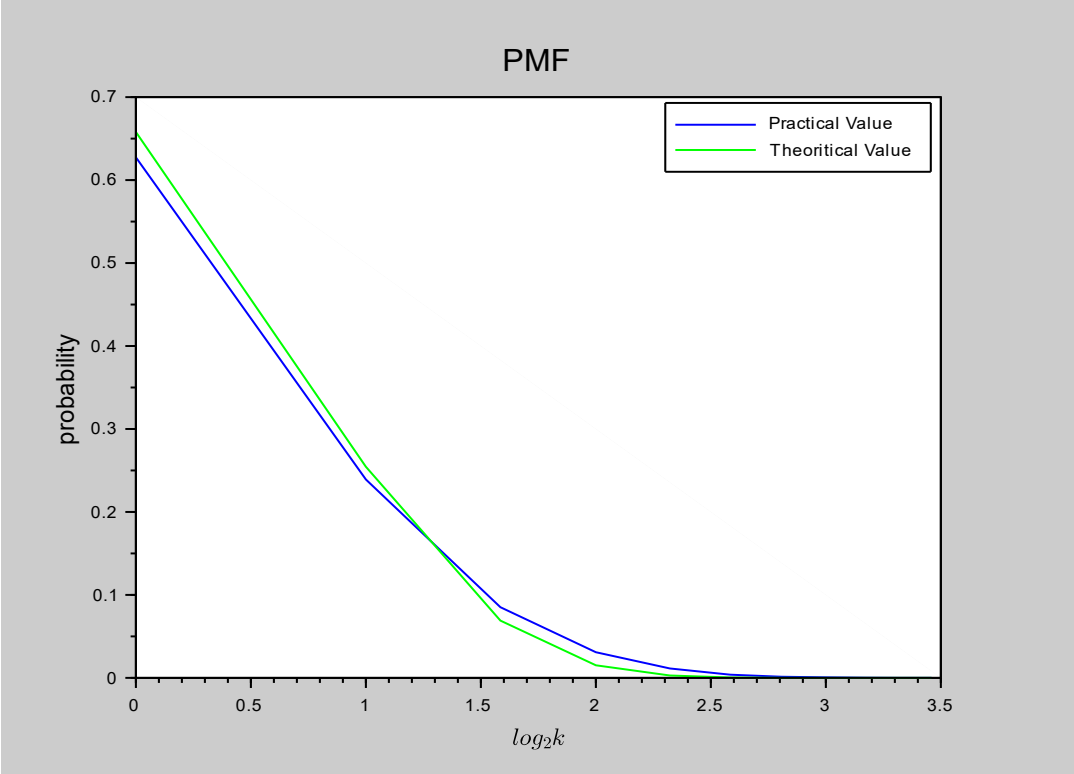

Figure 4.6: Comparison of Distributions (eT = 1, σP = 2, σS = 1 and μP = 0.01)

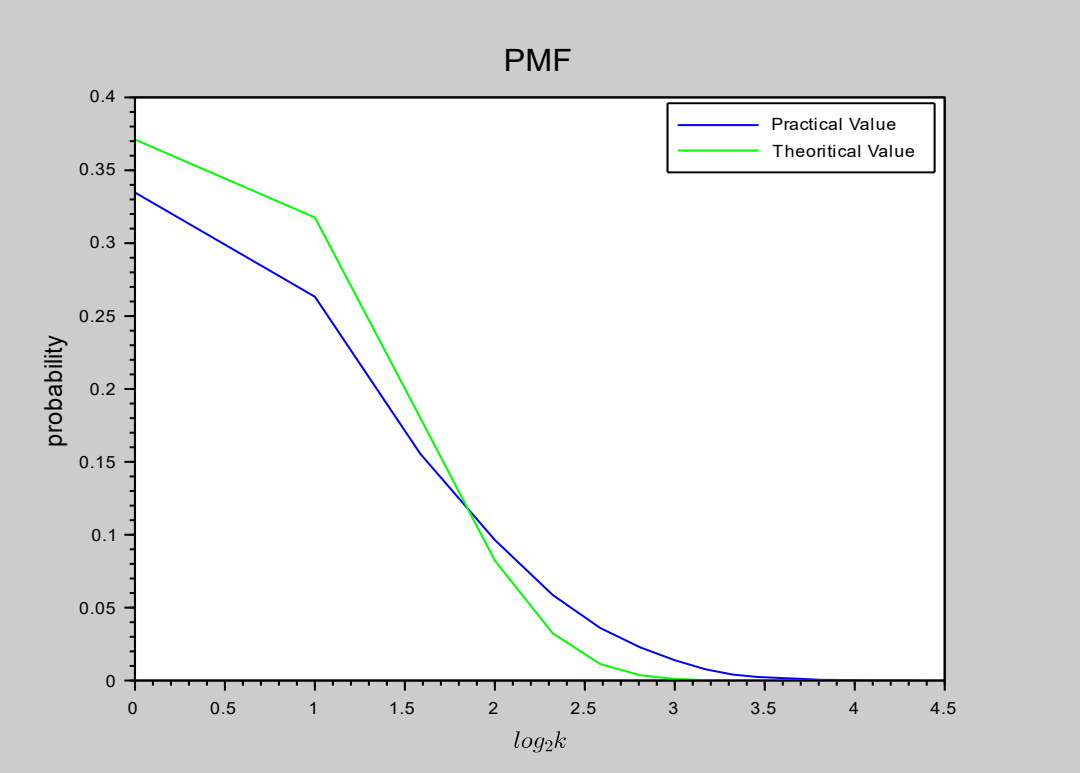

Figure 4.7: Comparison of Distributions ($e_T = 2$, $\sigma_P = 2$, $\sigma_S = 1$ and $\mu_P = 0.01$)

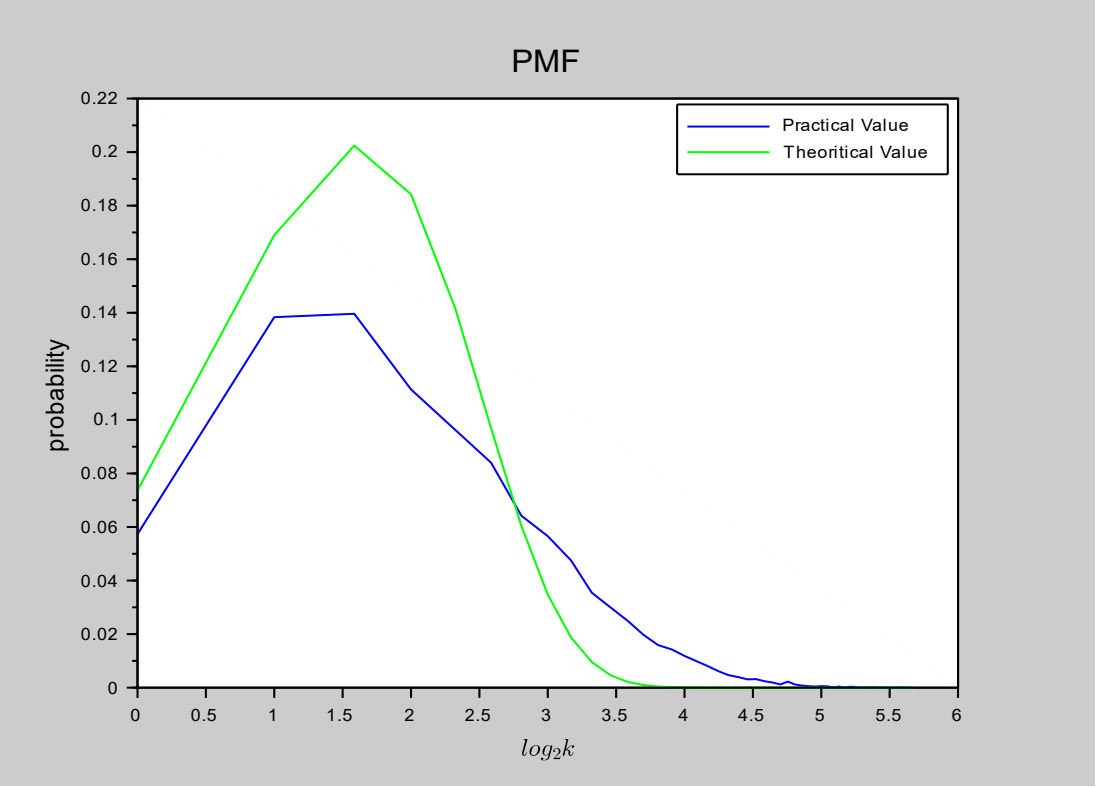

Figure 4.8: Comparison of Distributions ($e_T = 4$, $\sigma_P = 2$, $\sigma_S = 1$ and $\mu_P = 0.01$)

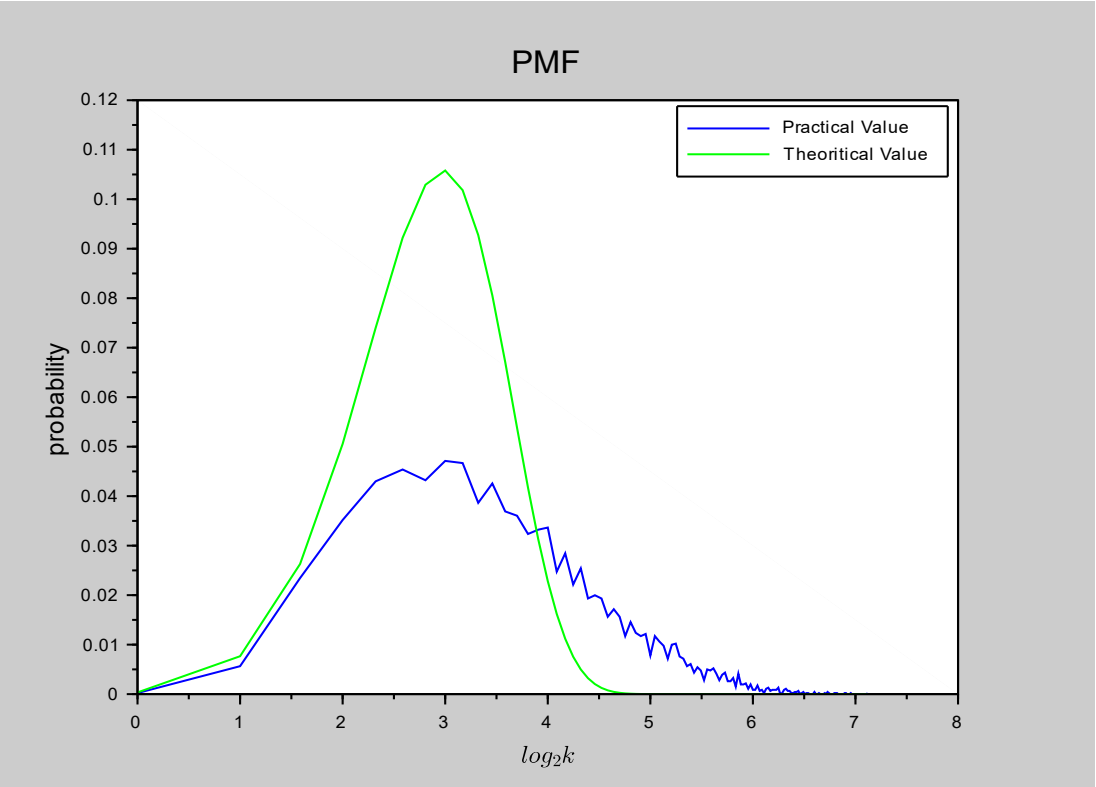

Figure 4.9: Comparison of Distributions ($e_T = 8$, $\sigma_P = 2$, $\sigma_S = 1$ and $\mu_P = 0.01$)

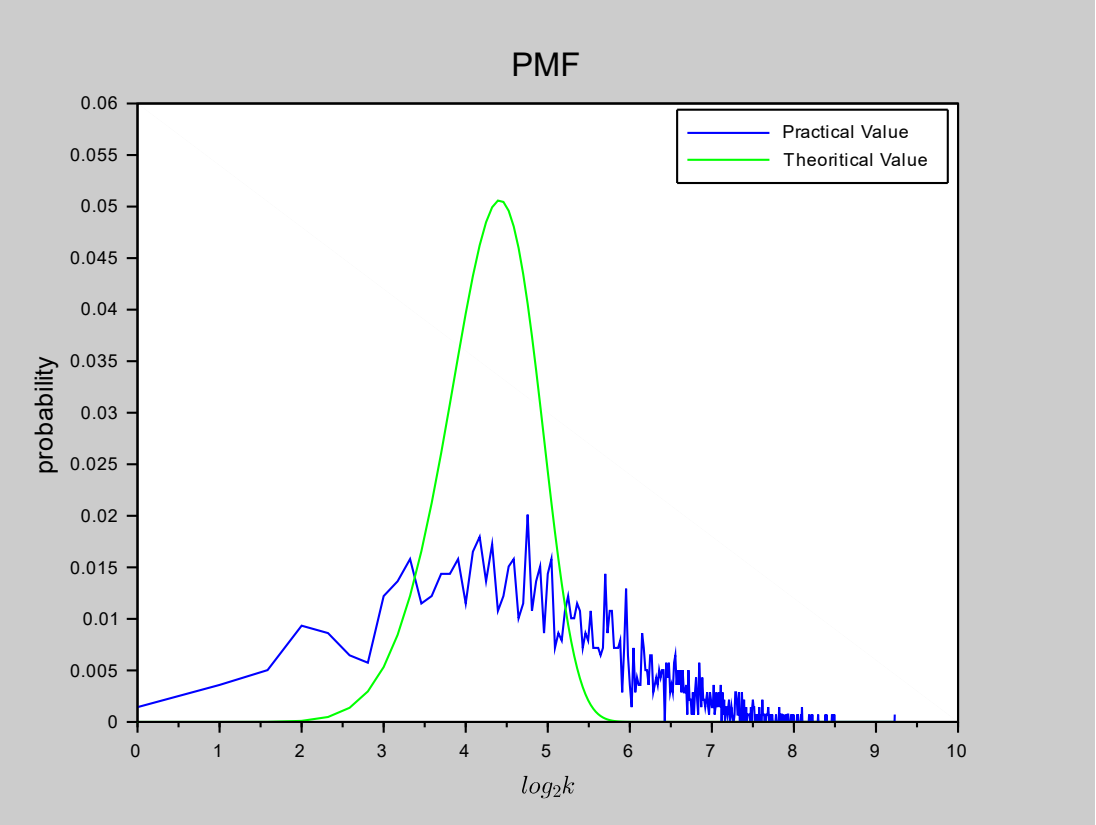

Figure 4.10: Comparison of Distributions (e_T = 16, σ_P = 2, σ_S = 1 and μ_P = 0.01)

In this experiment e_T value should be under 3.1723. We can clearly see when the e_T value is greater than that there is a clear difference between two distributions.

Simulation for σ_P = 1, σ_S = 1 and μ_P = 0.01

According to the equation 11, for values of σ_P = 1, σ_S = 1 and μ_P = 0.01, e_T should be less than 1.99. Here also we can clearly observe the deviation of graphs when the e_T value is not satisfying the equation 11. Figure 4.11, Figure 4.12, Figure 4.13, Figure 4.14 and Figure 4.15 show the Probability Mass Function for different e_T values.

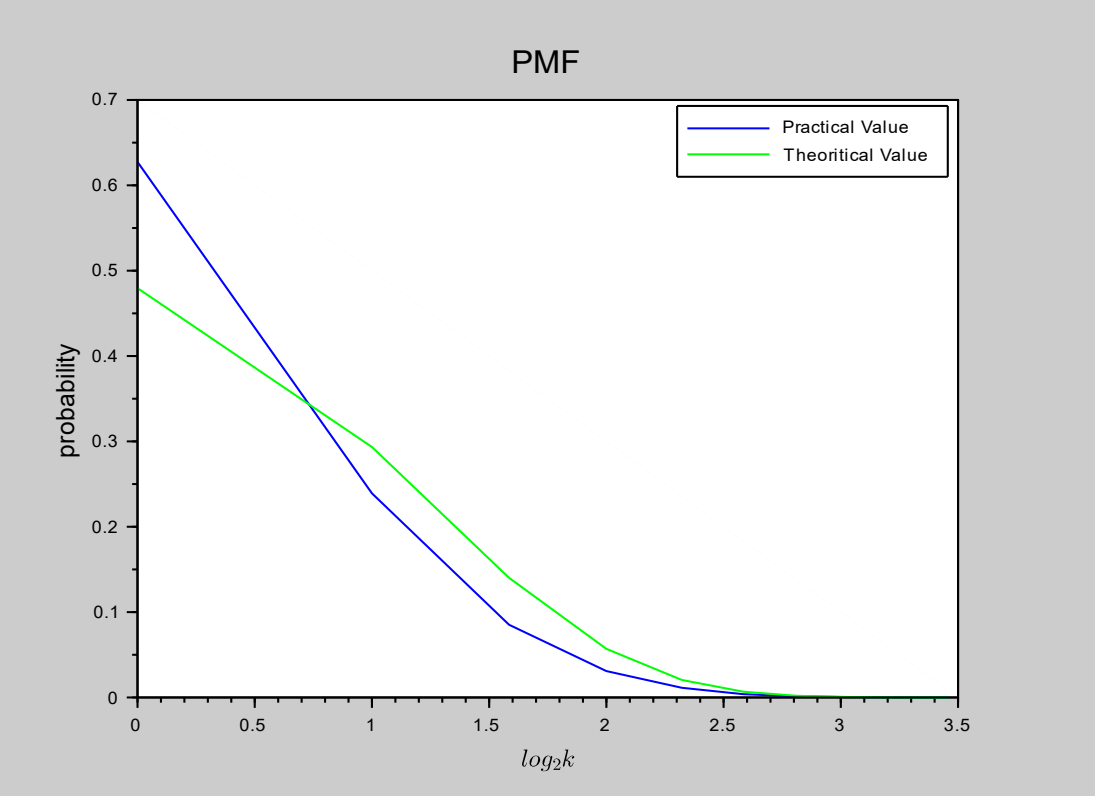

Figure 4.11: Comparison of Distributions ($e_T = 1$, $\sigma_P = 1$, $\sigma_S = 1$ and $\mu_P = 0.01$)

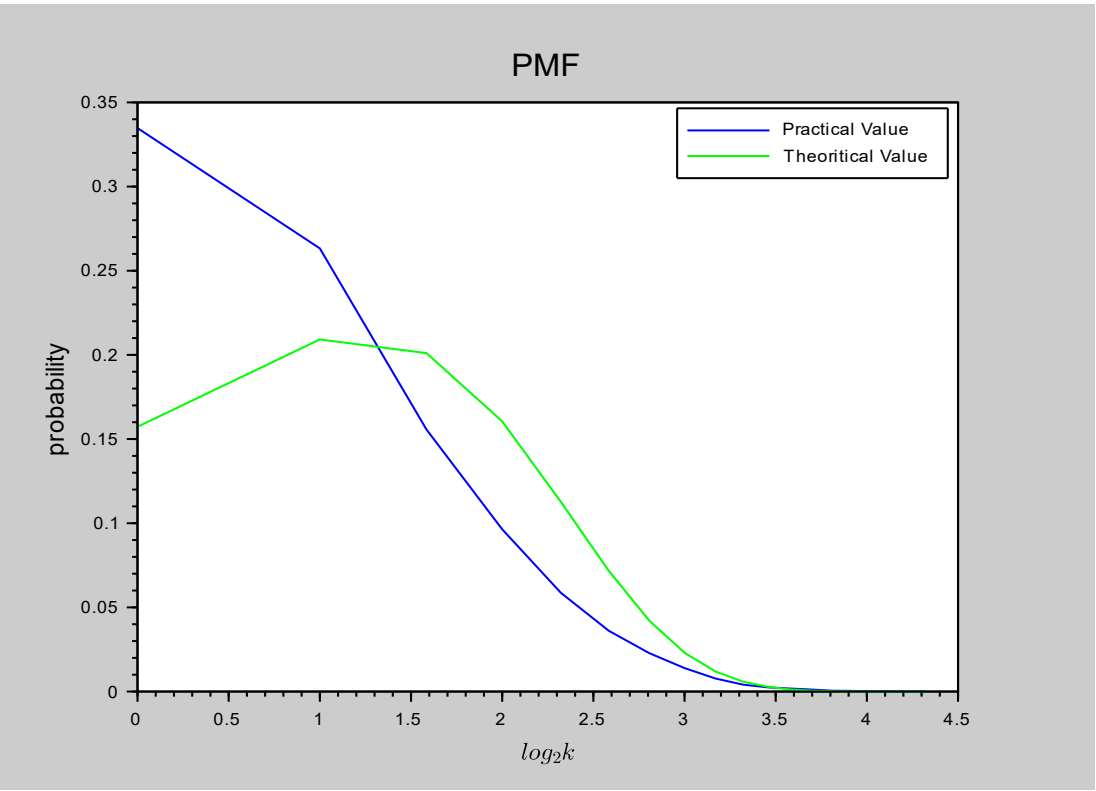

Figure 4.12: Comparison of Distributions ($e_T = 2$, $\sigma_P = 1$, $\sigma_S = 1$ and $\mu_P = 0.01$)

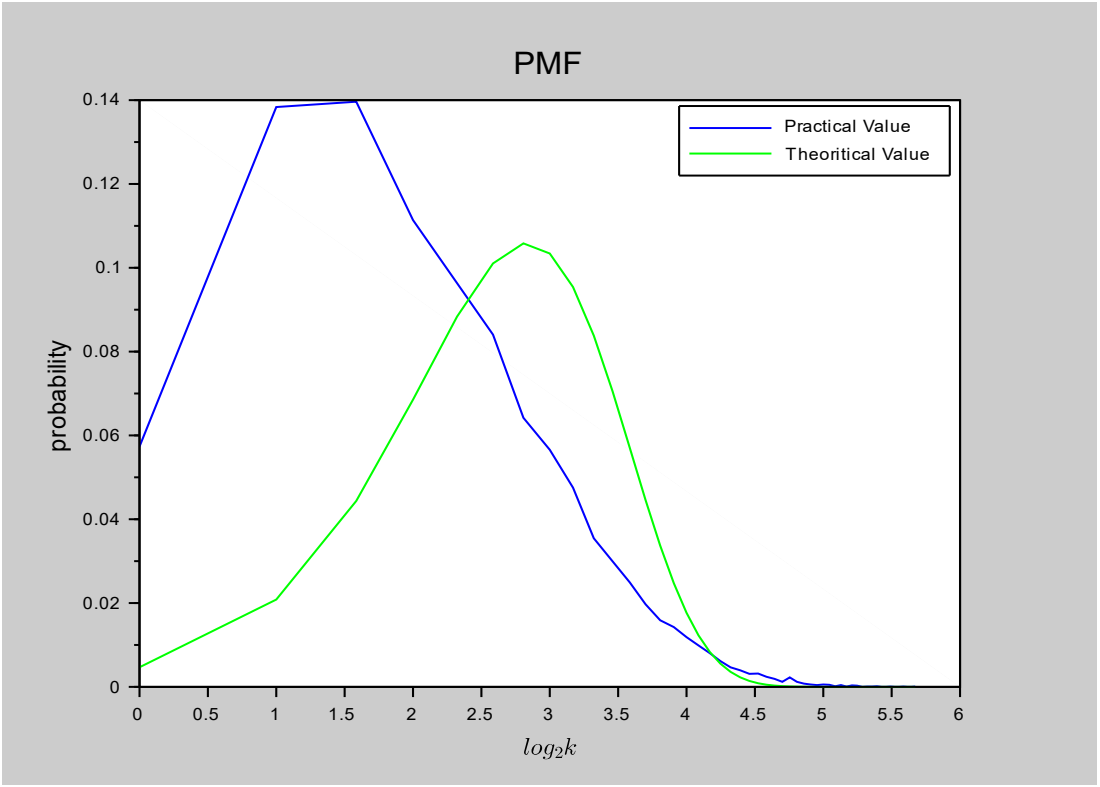

Figure 4.13: Comparison of Distributions ($e_T = 4$, $\sigma_P = 1$, $\sigma_S = 1$ and $\mu_P = 0.01$)

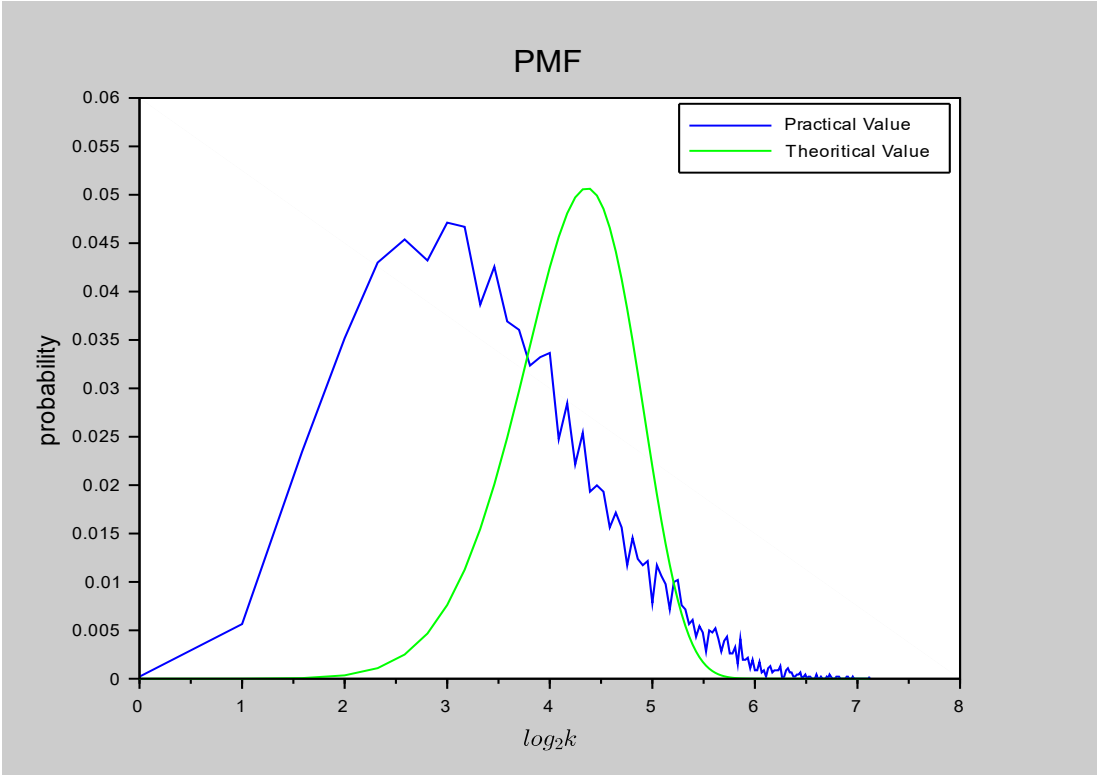

Figure 4.14: Comparison of Distributions ($e_T = 8$, $\sigma_P = 1$, $\sigma_S = 1$ and $\mu_P = 0.01$)

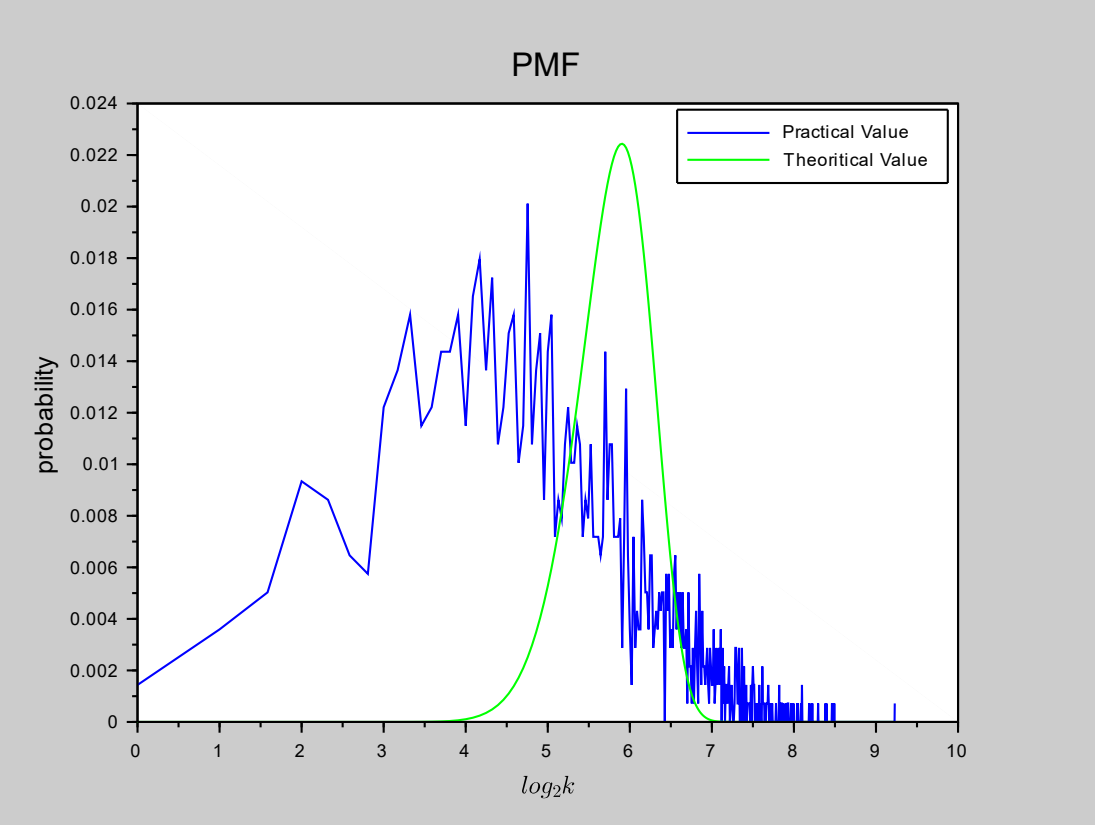

Figure 4.15: Comparison of Distributions (e_T = 16, σ_P = 1, σ_S = 1 and μ_P = 0.01)

As seen in the graphs, when the e_T value is large there is a difference between the theoretical value and the empirically calculated value. The reason is, Q function value are used to calculate the theoretical model. When the e_T is large, we are getting values from the tail of the Q function. When the Q function accuracy degrade, the theoretical model gives different values compared to the empirically calculated values. This will not be a problem because as in the equation 12, we are getting the one over value of the expected value of the function to calculate the average traffic rate and the theoretical model act as an upper bound.

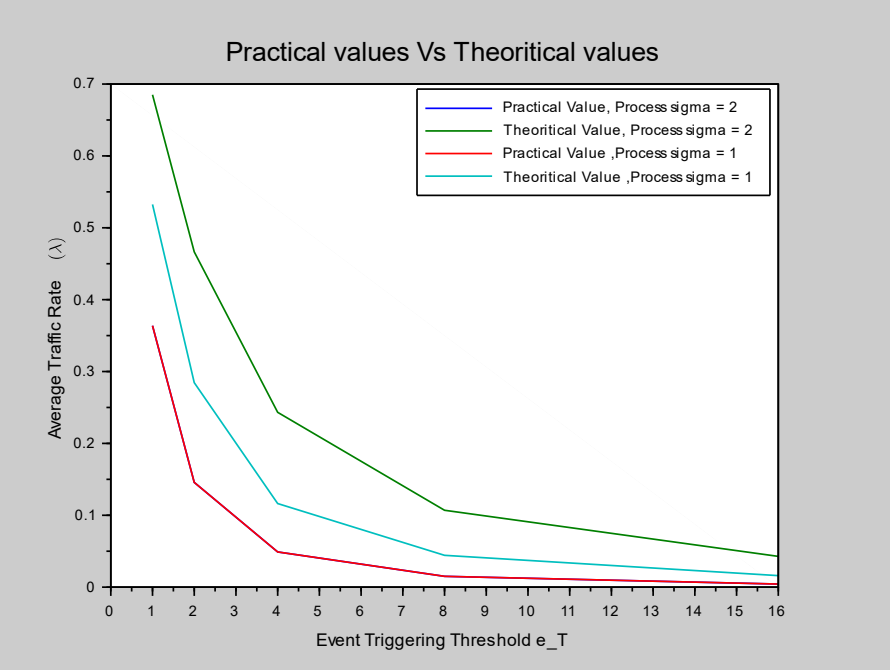

Figure 4.16: A Comparison Between the Empirical and Theoretical Values for Two Process Variances

Values in the Figure 4.16 was obtained for $\sigma_S = 1$ and $\mu_P = 0.01$. It's obvious that when the variance is low both the models follow a same pattern and when the variance is comparatively high, theoretical model acts as an upper bound.

4.2 Multiple Sensor Simulations

The phase two of the simulation done for $e_T = 1$, $\sigma_P = 1$, $\sigma_S = 0.5$ and $\mu_P = 0.01$.

Table 4.3: Analysis for Homogeneous Sensors

Number of Sensors	EQN 11 RHS	TVD	KLD	E[Prac]	E[Theo]	$\lambda_{Practical}$	$\lambda_{Theoritical}$
5	1.58	0.053	0.087	2.47	2.14	0.41	0.47
10	1.58	0.048	0.086	2.47	2.14	0.41	0.47
20	1.58	0.051	0.082	2.47	2.14	0.41	0.47
50	1.58	0.050	0.083	2.47	2.14	0.41	0.47
100	1.58	0.051	0.086	2.47	2.14	0.41	0.47

Simulations were done for sets of nodes with 5, 10, 20, 50 and 100. A standard Ethernet network can transmit data at a rate up to 10 Megabits per second (10 Mbps). Other LAN types include Token Ring, Fast Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet, Fiber Distributed Data Interface (FDDI). Fast Ethernet carry traffic at the nominal rate

of 100 Mbit/s. Fast ethernet was defined as the as the IEEE 802.3u standard in 1995. The standard Ethernet (IEEE 802.3) frame size is 1,518 bytes with additional frame overhead of 18 bytes. Minimum frame size of Ethernet in 64 Bytes. With the packet preamble of 7 Bytes and Frame Delimiter of 1 Byte, the total packet size would be 72 Bytes.

With equation 14 and 15, we can find the theoretical n_{max} which exceed the threshold tB for a given network bandwidth. When we consider a higher bandwidth, the value we get to n_{max} is too higher to simulate in OMNeT++. For simulation perpose only we are considering two small bandwidth values as 0.25 Mbps and 0.1 Mbps.

In OMNeT++ simulations were done for 9223300 seconds. Here only a part of the data (time 0 seconds to 100000 seconds) is shown due to the large volume of generated data.

Aggregated traffic values are presented here. t = 0.75 and t*B value should be divided by the packet size when applying to equation 15. Quality of service threshold $P_d = p(H_n > tB)$ is taken as 0.01

Simulation 1

B = 0.25 Mbps and then t*B = 0.1875 Mb. According to equation 15, n_{max} value is 771.

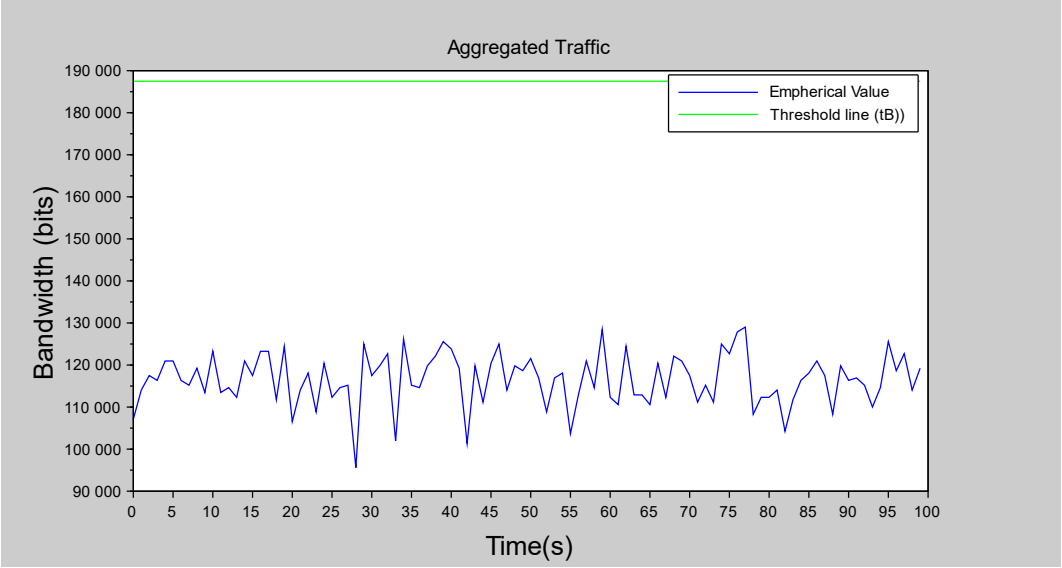

Figure 4.17: Aggregated Traffic when n = 500

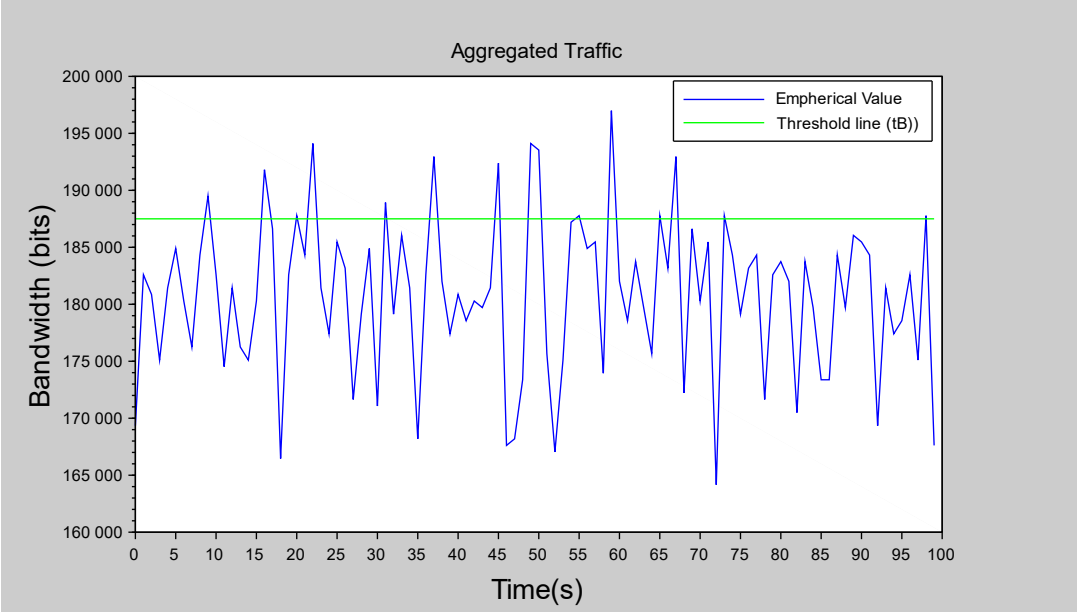

Figure 4.18: Aggregated Traffic when n = 771

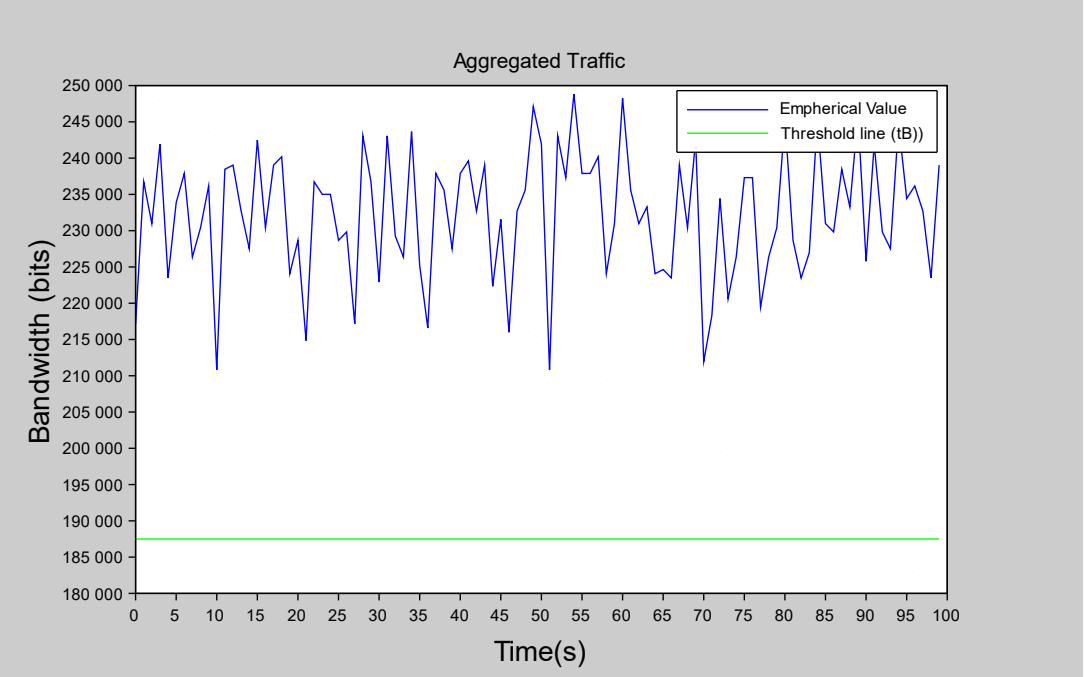

Figure 4.19: Aggregated Traffic when n = 1000

Simulation 2

B = 0.1 Mbps and then t*B = 0.075 Mb. According to equation 15, n_{max} value is 329.

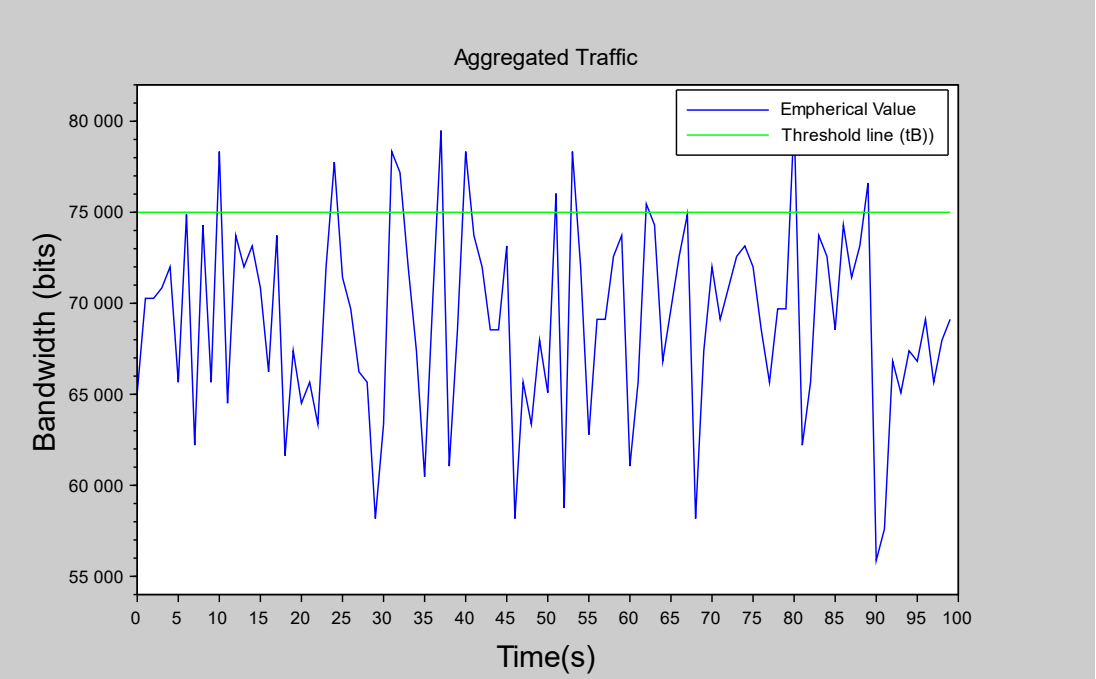

Figure 4.20: Aggregated Traffic when *n* = 300

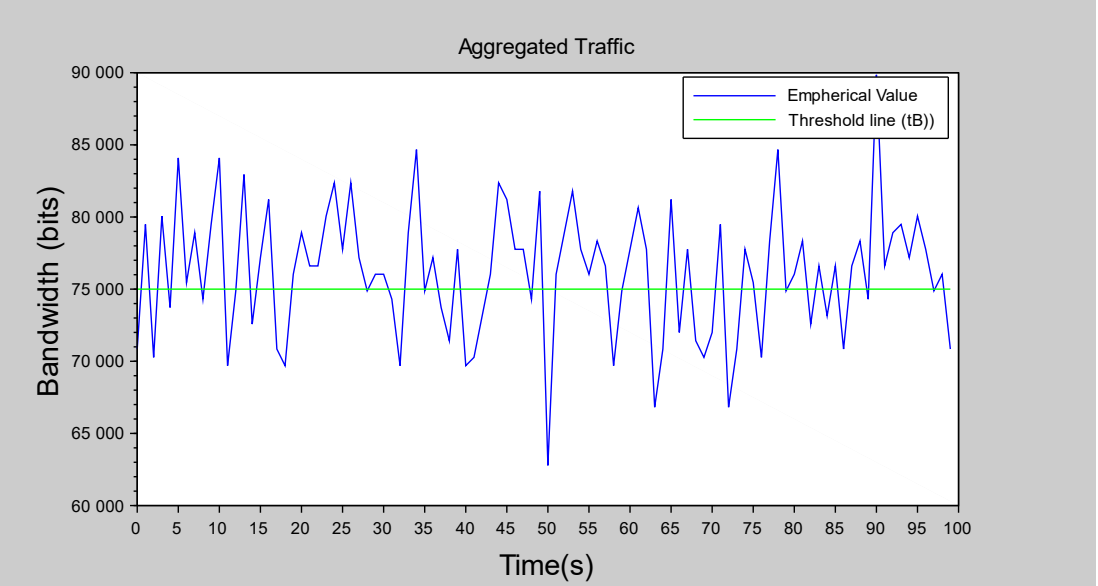

Figure 4.21: Aggregated Traffic when *n* = 329

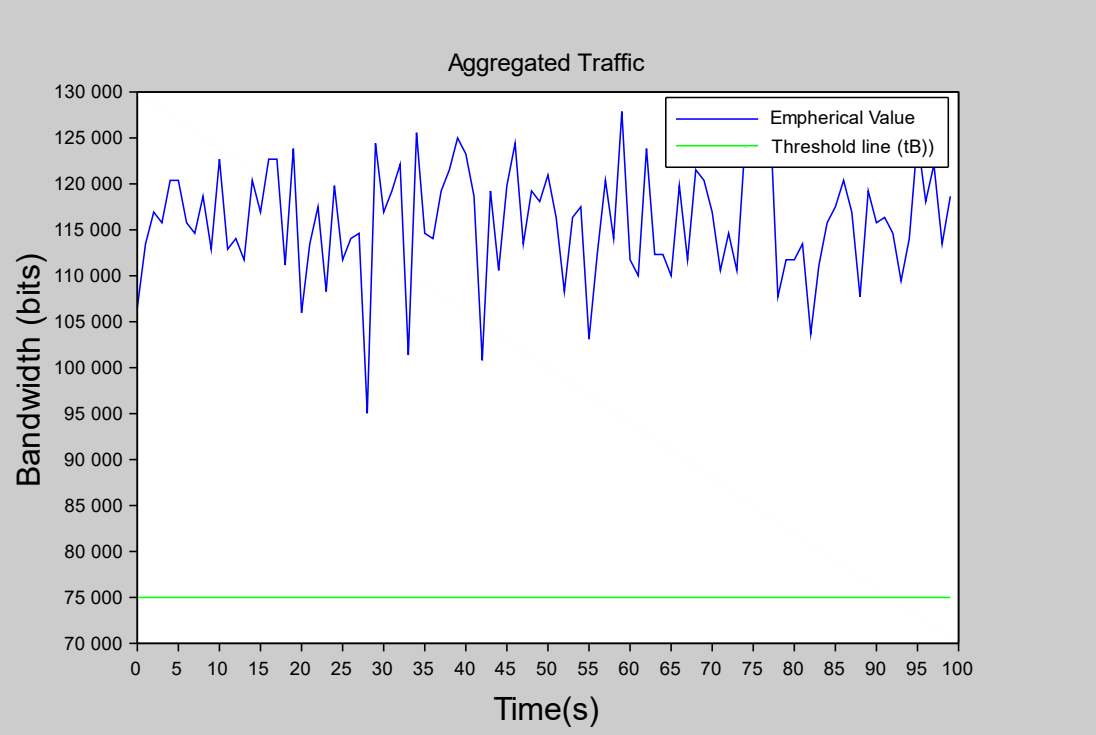

Figure 4.22: Aggregated Traffic when n = 500

When the bandwidth is set to 0.25 Mbits, the right hand side of the equation 14 gives a probability value 0.01008 for n = 771.When bandwidth and n are set to 0.1 Mbits and 329 respectively the right hand side of the equation 14 gives a probability value 0.01032.

CHAPTER 5

5. CONCLUSION AND FUTERE DIRECTIONS

A statistical model of the generated traffic is a must in order to effectively mitigate the congestion. From the model we obtained in the previous chapters we could easily determine the maximum number of sensors for a given network bandwidth and required quality of service to be determined. This information is vital in critical applications such as in the medical field because with this information, it will be easy to provide a high quality of service and predict the network behaviour. From the empirical results we obtained from OMNeT++ simulations, it's obvious that the model is correct. When the e_T values are smaller than the right-hand side of the equation 11, PMF of the empirical results and theoretical model is almost identical. Due to the low accuracies of the tail values of Q function, a relatively larger mean value is obtained for the empirical mean. It will not affect in average traffic calculations because we are using the one over value of the mean and is used to get an upper bound threshold value.

In Multiple Sensor Simulations 4.2, simulation 1 and 2 shows that the value getting for the equation 15 is in the correct zone. In both cases the resultant values satisfy the inequality. In both cases, when n is set to a low number compared to the n_{max} value, the equation 14 gives very low value as the probability. When n is set to a large number compared to the n_{max} the equation 14 gives a very low probability value. As per the Figure *4.17* and Figure *4.19* resultant values verify the developed model.

As a next step we can model and simulate a network with heterogeneous sensors. In the heterogeneous case, the difference in input process, sensor and encoder parameters will result in $p(V_i = 1) = \lambda_i$. This will result in (13) becoming Poisson Binomial for small n. Developing a model to get n_{max} in a network with heterogeneous sensors will be an interesting future direction. Further research will be done to compare the $p(H_n > t * B)$ and $t*B$ values. A work flow is already being developed for this.

6. APPENDIX 1

Additional Results

A network with several sensors were created in OMNeT++ and while keeping the σ_P, σ_S, μ_P values same, several simulations were done by changing the e_T value to examine the behaviour of the λ_{Prac} and λ_{Theo} values. Few of the graphs will be presented here for a better understanding of the model behaviour. Here y axis is λ values and x axis is e_T values.

n = 10

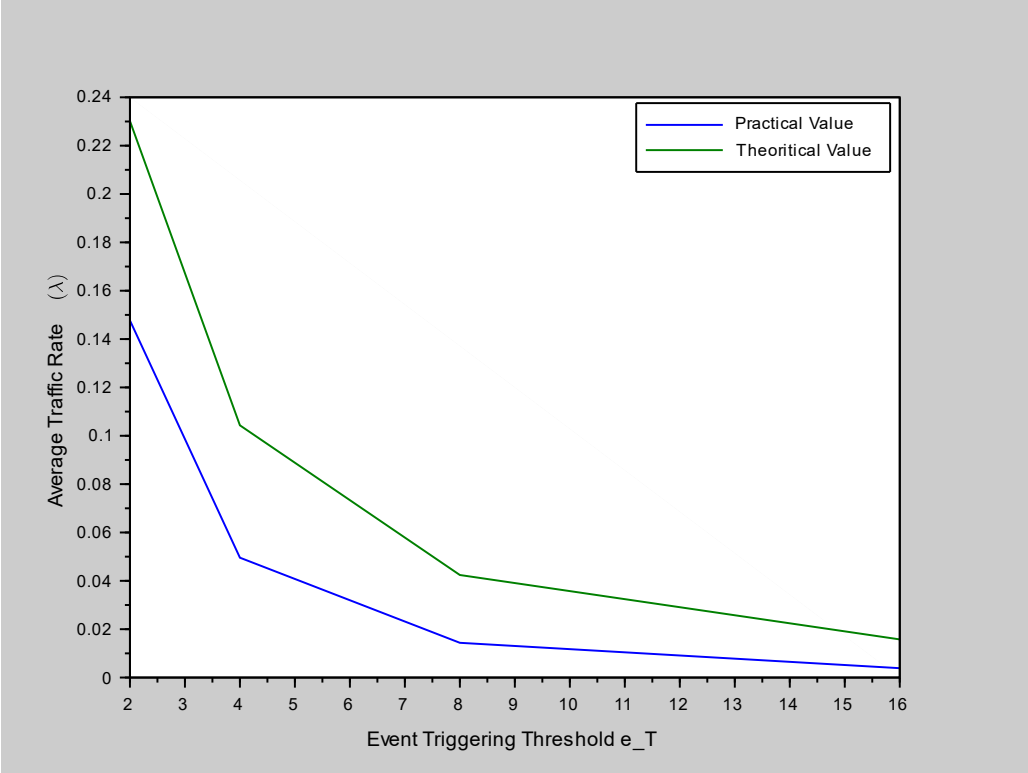

Figure 6.1: Comparison of Distributions (σ_P = 1, σ_S = 0.1 and μ_P = 0.01)

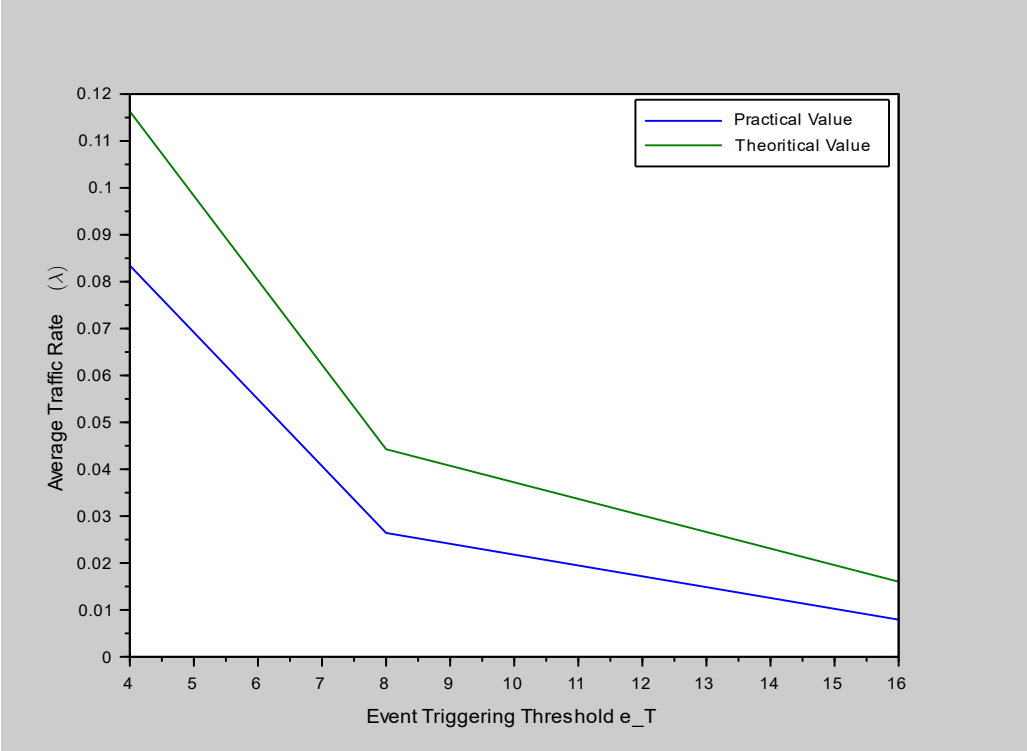

Figure 6.2: Comparison of Distributions ($\sigma_P = 1$, $\sigma_S = 1$ and $\mu_P = 0.01$)

$n = 100$

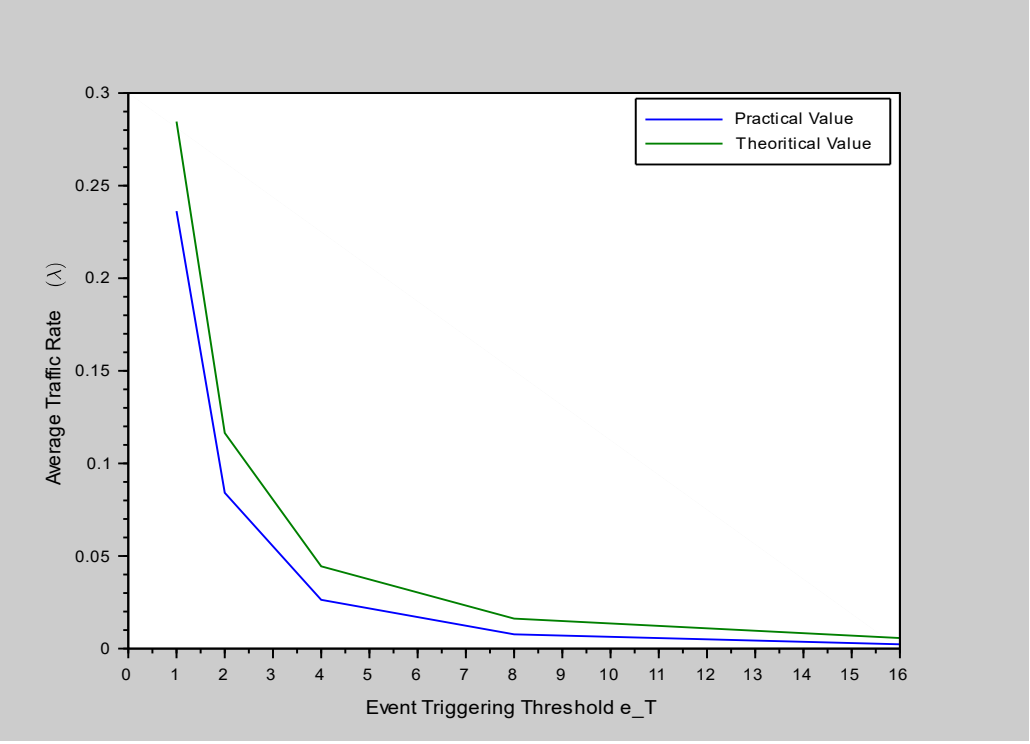

Figure 6.3: Comparison of Distributions ($\sigma_P = 0.5$, $\sigma_S = 0.5$ and $\mu_P = 0.01$)

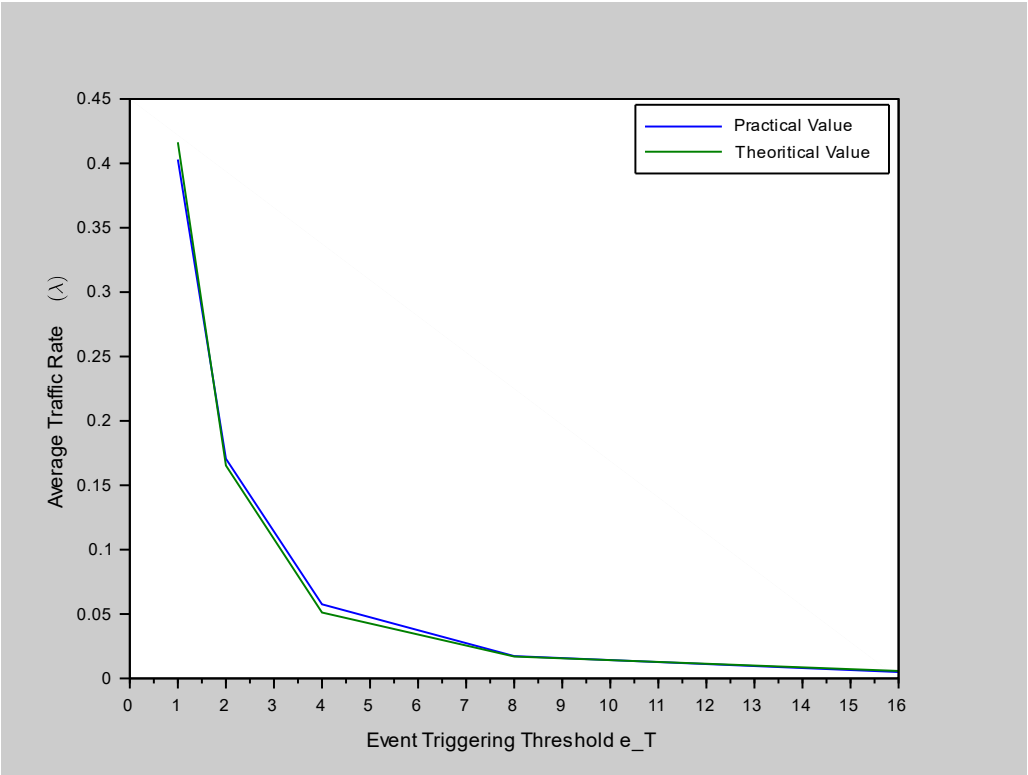

Figure 6.4: Comparison of Distributions ($\sigma_P = 0.5$, $\sigma_S = 1$ and $\mu_P = 0.01$)

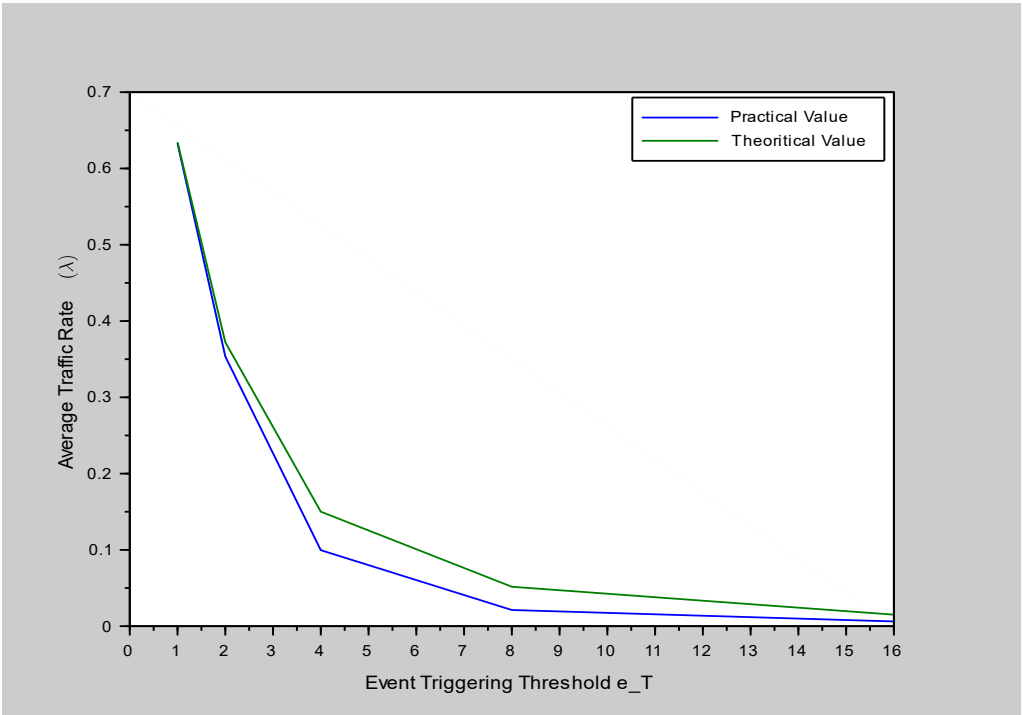

Figure 6.5: Comparison of Distributions ($\sigma_P = 0.5$, $\sigma_S = 2$ and $\mu_P = 0.01$)

7. REFERENCES

[1] S. Gupta and S. Misra, "Moderating effect of compliance, network, and security on the critical success factors in the implementation of cloud ERP", *IEEE Transactions on Cloud Computing*, vol. 4, no. 4, pp. 440-451, 2016.

[2] J. Antes and I. Kallfass, “Performance estimation for broadband multi gigabit millimeter- and sub-millimeter-wave wireless communication links”, *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 10, pp. 3288–3299, 2015.

[3] J. M. Khurpade, D. Rao and P. D. Sanghavi, "A survey on IoT and 5G network", *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, pp. 1-3, 2018.

[4] Q. Liu, Z. Wang, X. He and D. Zhou, "A survey of event-based strategies on control and estimation", *Systems Science and Control Engineering*, vol. 2, no. 1, pp. 90-97, 2014.

[5] A. Cervin, K. J. Åström, “On limit cycles in event-based control systems”, *IEEE Conference on Decision and Control*, pp. 1947-1952, 2007.

[6] K. Åarzén, "A simple event-based PID controller", *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 8687-8692, 1999.

[7] U. Premaratne, S. Halgamuge and I. Mareels, "Event triggered adaptive differential modulation: a new method for traffic reduction in networked control systems", *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1696-1706, 2013.

[8] U. Premaratne, S. K. Halgamuge, and I. M. Y. Mareels, “Traffic reduction in packet switched networked control systems using deadband error modulation”, *IEEE Transactions on Automatic Control*, Vol. 62, no.8, pp.4038 – 4043, 2017.

[9] U. Premaratne, S. Warnakulasooriya and R. Nandana, "Characterization of event-based sampling encoders for Industrial Internet of Things using input–output mutual information", *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5495-5505, 2021.

[10] E. Priyanka, S. Thangavel and X. Gao, "Review analysis on cloud computing based smart grid technology in the oil pipeline sensor network system", *Petroleum Research*, vol. 6, no. 1, pp. 77-90, 2021.

[11] U. Premaratne, "Limit cycle like asymptotic dynamics in feedback loops with memory based event triggering", *2016 Moratuwa Engineering Research Conference (MERCon)*, Moratuwa, pp. 427-431, 2016.

[12] S. A. Busari, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, “Terahertz massive MIMO for beyond 5G wireless communication”, *IEEE International Conference on Communications (ICC)*. IEEE, pp. 1–6, 2019.

[13] C. Bass, “Local area networks-a merger of computer and communications technologies”, *Microprocessors and Microsystems*, vol. 5, no. 5, pp. 187–192, 1981.

[14] L. Shu, Y. Chen, Z. Huo, N. Bergmann, and L. Wang, “When mobile crowd sensing meets traditional industry”, *IEEE Access*, vol. 5, pp. 15 300–15 307, 2017.

[15] C. M. Park, R. A. Rehman, and B.-S. Kim, “Packet flooding mitigation in CNN-based wireless multimedia sensor networks for smart cities”, *IEEE Access*, vol. 5, pp. 11 054–11 062, 2017.

[16] C. Bejaoui, A. Guitton and A. Kachouri, "Equal size clusters to reduce congestion in wireless multimedia sensor Nntworks", *Wireless Personal Communications*, vol. 97, no. 3, pp. 3465-3482, 2017.

[17] M. Alaei, P. Sabbagh, and F. Yazdanpanah, “A QoS-aware congestion control mechanism for wireless multimedia sensor networks”, *Wireless Networks*, vol. 25, no. 7, pp. 4173–4192, 2019.

[18] K. Sumathi and P. Pandiaraja, “Dynamic alternate buffer switching and congestion control in wireless multimedia sensor networks”, *Peer-to-Peer Networking and Applications*, vol. 13, no. 6, pp. 2001-2010, 2019.

[19] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial Internet of Things: challenges, opportunities, and directions”, *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[20] F. Nizzi, T. Pecorella, M. Bertini, R. Fantacci, M. Bastianini, C. Cerboni, A. Buzzigoli, M. Gattoni, and A. Fratini, “Evaluation of IoT and video surveillance applications in a 5G smart city: the Italian 5G experimentation in Prato”, *AEIT International Annual Conference. IEEE*, pp. 1–6, 2018.

[21] E. Sultanow and A. Chircu, “A review of IoT technologies, standards, tools, frameworks and platforms”, in *The Internet of Things in the Industrial Sector. Springer*, pp. 3–34, 2019.

[22] C. Lee, C. Park, K. Jang, S. Moon, and D. Han, “DX: latency based congestion control for datacenters”, *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 335–348, 2016.

[23] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, F. Kelly, M. Alizadeh and M. Yu, “HPCC: high precision congestion control”, *ACM Special Interest Group on Data Communication*, pp. 44–58, 2019.

[24] H. Wu, Z. Feng, C. Guo, and Y. Zhang, “ICTCP: incast congestion control for TCP in data-center networks”, *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 345–358, 2013.

[25] H. Rezaei, M. U. Chaudhry, H. Almasi, and B. Vamanan, “ICON: incast congestion control using packet pacing in datacenter networks”, *11th International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, pp. 125–132, 2019.

[26] M. Gowtham and K. Subramaniam, “Congestion control and packet recovery for cross layer approach in manet”, *Cluster Computing*, vol. 22, no. 5, pp. 12 029–12 036, 2019.

[27] A. Zhou, S. Wang, Z. Zheng, C.-H. Hsu, M. R. Lyu, and F. Yang, “On cloud service reliability enhancement with optimal resource usage”, *IEEE Transactions on Cloud Computing*, vol. 4, no. 4, pp. 452–466, 2014.

[28] I. Kim, J. Hwang, W. Wang and M. Humphrey, "Guaranteeing performance SLAs of cloud applications under resource storms", *IEEE Transactions on Cloud Computing*, pp. 1-1, 2020.

[29] M. Hu, D. Wu, W. Wu, J. Cheng and M. Chen, "Quantifying the influence of intermittent connectivity on mobile edge computing", *IEEE Transactions on Cloud Computing*, pp. 1-1, 2019.

[30] Y. Lu, X. Huang, K. Zhang, S. Maharjan and Y. Zhang, "Communication-efficient federated learning for digital twin edge networks in Industrial IoT", *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5709-5718, 2021.

[31] G. C. Walsh, H. Ye, and L. G. Bushnell, “Stability analysis of networked control systems”, *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438–446, 2002.

[32] A. E. Abdelaal, T. Hegazy, and M. Hefeeda, “Event-based control as a cloud service”, *American Control Conference*, 2017. IEEE, pp. 1017–1023, 2017.

[33] C. Canudas-de-Wit, F. Gomez-Estern, and F. R. Rubio, “Delta modulation coding redesign for feedback controlled systems”, *IEEE Transactions on Industrial Electronics*, vol. 56, no. 7, pp. 1–20, 2009.

[34] F. Gomez-Estern, C. Canudas-de-Wit, and F. R. Rubio, “Adaptive delta modulation in networked controlled systems with bounded disturbances”, *IEEE Transaction on Automatic Control*, vol. 56, no. 1, pp. 129–134, 2011.

[35] D. Almakhles, A. K. Swain, A. Nasiri, and N. Patel, “An adaptive two-level quantizer for networked control systems”, *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 1084–1091, 2016.

[36] N. Akhtar, M. A. Khan, A. Ullah, and M. Y. Javed, “Congestion avoidance for smart devices by caching information in manets and iot”, *IEEE Access*, vol. 7, pp. 71 459–71 471, 2019.

[37] C. Zhang, C. Xia, Y. Li, H. Wang, and X. Li, “A hotspot-based probabilistic cache placement policy for ICN in MANETs”, *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 134, 2019.

[38] V. Wijesinghe and U. Premaratne, "Bandwidth reduction and convergence analysis of extremum seeking control with feedback encoding", *Frontiers in Mechanical Engineering*, vol. 2, 2016.

[39] M. V. Ramesh, “Design, development, and deployment of a wireless sensor network for detection of landslides”, *Ad Hoc Networks*, vol. 13, pp. 2–18, 2014.

[40] M. Chiani, D. Dardari, and M. K. Simon, “New exponential bounds and approximations for the computation of error probability in fading channels”, *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 840–845, 2003.

[41] U. Premaratne, “Limit cycle like asymptotic dynamics in feedback loops with memory based event triggering”, in *2016 Moratuwa Engineering Research Conference*, pp. 427–431, 2016.

[42] M. Emara, H. Elsawy. and G. Bauch, “A spatiotemporal model for peak AoI in uplink IoT networks: time versus event-triggered traffic”, *IEEE Internet of Things Journal*, vol. 7, no. 8, pp.6762-6777, 2020.

[43] B. Mohebali, A. Tahmassebi, A. Gandomi, and A. Meyer-Baese "A big data inspired preprocessing scheme for bandwidth use optimization in smart cities applications using Raspberry Pi", *SPIE Big Data: Learning, Analytics, and Applications*, vol. 10989, 2019.

[44] M. Samimi. and T. Rappaport, “3-D millimeter-wave statistical channel model for 5G wireless system design”, *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 7, pp.2207-2225, 2016.

[45] R. Nandana, “Comparative study of traffic reduction strategies in networked control systems”, Master of Science thesis, University of Moratuwa, 2018.