12-2013

Client-Based QoS Monitoring and Evaluation Architecture for Network Infrastructure and Services

Ammar Mohammed Kamel

Western Michigan University, AmarMK2003@yahoo.com

Follow this and additional works at: http://scholarworks.wmich.edu/dissertations

Part of the Databases and Information Systems Commons, and the OS and Networks Commons

Recommended Citation

http://scholarworks.wmich.edu/dissertations/210

This Dissertation-Open Access is brought to you for free and open access by the Graduate College at ScholarWorks at WMU. It has been accepted for inclusion in Dissertations by an authorized administrator of ScholarWorks at WMU. For more information, please contact maira.bundza@wmich.edu.
CLIENT-BASED QOS MONITORING AND EVALUATION ARCHITECTURE
FOR NETWORK INFRASTRUCTURE AND SERVICES

by

Ammar Mohammed Kamel

A Dissertation Submitted to the Graduate College
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
Computer Science
Western Michigan University
December 2013

Doctoral Committee:

Ala Al-Fuqaha, Ph.D., Chair
Dionysios Kountanis, Ph.D.
Ammar Rayes, Ph.D.
Driss Benhaddou, Ph.D.
Providing an efficient Quality-of-Service (QoS) measurement model is a challenging problem in today’s mobile computing and telecommunications networks. Currently, most of QoS techniques utilize service measurements that are collected by the network elements (i.e., network-side monitoring) to evaluate the network performance. However, this process does not take into account the service performance from the clients' perspective and might contradict with the Service Level Agreement (SLA). In order to overcome the limitations of service-side QoS monitoring, a number of research studies have been conducted to present alternative architectures and algorithms for client-side QoS service assessment in computer networks. The client-side QoS approach gives the major role to the clients to evaluate the dedicated services through gathering the network measurements and reporting the necessary information. The service providers consider clients’ feedback to revise and resolve service performance issues. This model can be considered as a cooperative approach that provides a compromised service plan for both the service providers and the network clients to achieve a better network service performance.
In this research, we study the tradeoffs between network-side and client-side QoS monitoring and present a client-based architecture for the evaluation and prediction of service degradations in mobile networks. The client-side approach should be capable of utilizing multi-level service performance analysis in a scalable mobile network environment. The service performance analysis consists of three levels: service monitoring and evaluation, verifying and enhancing the network measurement accuracy (the collected performance data), and performance prediction in single and multi-hops networks. In addition, this approach should support short and long-term service evaluation scenarios. The short-term scenario gives the ability to service providers to react in a timely manner to the clients' feedback to preserve a certain level of service performance. The long-term scenario helps to clarify the service behavior by predicting the service degradation over the monitoring and evaluation sessions. Furthermore, this scenario allows the service providers to refine the degraded services and maintain the SLA.
ACKNOWLEDGMENTS

First and foremost, I would like to show the great appreciation to God for the continued support and guidance in this life, and without his willing I would not be reached this stage of the success. Also, I wish to express my sincere gratitude to Dr. Ala Al-Fuqaha for providing me with encouragement, assistance, invaluable knowledge and experience. Special thanks to my committee members Dr. Dionysios Kountanis, Dr. Ammar Rayes, and Dr. Driss Benhaddou for their important suggestions, comments and supervision that lead me to the right direction to finish this work.

I want also to thank my parents for their support and prayers. They were and still are the light which shines in my life. Lastly, I would like to thank my wife who sincerely devoted her time and effort to make me walk through this stage of my life. To all of you and to my lovely son, Yassir, I am dedicating this work.

Ammar Mohammed Kamel
# TABLE OF CONTENTS

ACKNOWLEDGMENTS ........................................................................................................ ii

LIST OF TABLES ............................................................................................................... vii

LIST OF FIGURES ........................................................................................................ viii

CHAPTER

1. INTRODUCTION ....................................................................................................... 1

   1.1 Background and General Overview ......................................................................... 1

   1.2 Problem Statement .................................................................................................. 4

   1.3 Research Goals ...................................................................................................... 6

   1.4 Dissertation Outline ............................................................................................. 8

2. LITERATURE REVIEW ............................................................................................. 11

   2.1 General Overview .................................................................................................. 11

   2.2 QoS and Cloud Computing ................................................................................... 20

   2.3 QoS and Social Network Analysis .......................................................................... 22

3. CLIENT-BASED QOS SYSTEM DESIGN AND MODELING ................................ 25

   3.1 Introduction ............................................................................................................. 25

   3.2 Extreme Value Theorem (EVT) ............................................................................. 25

      3.2.1 GEV ............................................................................................................. 26

      3.2.2 GPD ............................................................................................................. 27

      3.2.2.1 Mean Residual Life Plot (MRL) ................................................................. 29

      3.2.2.2 Parameters Estimation against Thresholds (PET) ..................................... 29

iii
Table of Contents—continued

CHAPTER

3.3 Client-Based QoS Service Monitoring Assessment System ........ 31

3.4 Verifying the Efficiency of The proposed CBSMA System
   Architecture .......................................................................................... 35

3.4.1 Quality of Service Evaluation Algorithm (QoS-EA) ............ 35

3.5 Simulation Results .......................................................................... 40

3.6 Experimental Results and Analysis ............................................ 43

4. A CLIENT-BASED QOS DATA APPROACH USING
   GENERALIZED EXTREME VALUE THEOREM AND LINEAR
   OPINION POOL .................................................................................... 47

   4.1 Introduction .................................................................................. 47

   4.2 Linear Opinion Pool (LOP) .......................................................... 47

   4.3 Proposed Approach ..................................................................... 48

   4.3.1 Behavioral Assessment LOP Algorithm (BALOPA) ........... 50

   4.3.2 Mobile Clients Fidelity Algorithm (MCFA) ....................... 52

   4.4 Experimental Results ................................................................. 54

5. A CLIENT-BASED QOS APPROACH USING GENERALIZED
   EXTREME VALUE THEOREM IN MULTI-HOP NETWORK
   ENVIRONMENTS ............................................................................... 58

   5.1 Introduction .................................................................................. 58

   5.2 Problem Description and Modeling ........................................... 59

   5.3. The Proposed Algorithms ....................................................... 61

   5.3.1 Link Delay-Driven Algorithm (LDDA) .............................. 61
## Table of Contents—continued

### CHAPTER

5.3.2 Link Delay Performance Assessment Algorithm (LDPAA) ............................................................................. 63

5.4 Experimental Results .......................................................................................................................... 64

6. TOWARDS A CLIENT-SIDE QOS MONITORING AND ASSESSMENT USING GENERALIZED PARETO DISTRIBUTION IN A CLOUD-BASED ENVIRONMENT .................... 71

   6.1 Introduction ................................................................................................................................. 71

   6.2 The Proposed Approach ................................................................................................. 72

      6.2.1 On-Time Cloud Service Assessment Algorithm (OTCSA) ........................................ 76

   6.3 Experimental Results .............................................................................................................. 79

7. AN ADAPTIVE CLIENT-SIDE QOS SERVICE ASSESSMENT APPROACH USING SOCIAL NETWORK ANALYSIS AND EXTREME VALUE THEOREM ............................................................... 86

   7.1 Introduction ................................................................................................................................ 86

   7.2 Social Network Analysis (SNA) .............................................................................................. 87

   7.3 The Proposed Approach ............................................................................................................ 88

      7.3.1 Problem Description and Formulation ............................................................................... 89

      7.3.2 Extreme Social Bond Clustering Heuristic (ESBCH) ...................................................... 91

      7.3.3 Immediate Service Performance Assessment Algorithm (iSPA) .................................. 93

   7.4 Experimental Results .............................................................................................................. 94

      7.4.1 Methodology ....................................................................................................................... 94

      7.4.2 Results ............................................................................................................................... 95
<table>
<thead>
<tr>
<th>8. CONCLUSIONS AND FUTURE WORK</th>
<th>105</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Conclusions</td>
<td>105</td>
</tr>
<tr>
<td>8.2 Research Contributions</td>
<td>106</td>
</tr>
<tr>
<td>8.3 Future Work</td>
<td>109</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>111</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

3.1: (a) SMM in a Normal Form and (b) SMM in a Categorized Form ............... 39

3.2: The Selected GEV Model with Shape ($K$), Scale ($\sigma$), and Location ($\mu$) Parameters, For 25, 50, and 100 Clients.......................................................... 42

4.1: Server Performance Table (SPT)................................................................. 49

5.1: Estimated Links’ Delays of the Captured Snapshots.................................. 66

5.2: Sample of Estimated Links’ Delays of Single and Multi-Hop Paths .......... 66

5.3: All Possible Routes of MC$_5$........................................................................ 69

7.1: Goodness of Fit (Chi-Squared) Test for 105 Mobile Clients ....................... 96

7.2: Relative Frequencies and GPD Probabilities of the General and Cluster 1 Models............................................................................................................. 99

7.3: Relative Frequencies and GPD Probabilities of the General and Cluster 2 Models............................................................................................................. 99

7.4: Goodness of Fit (Chi-Squared) Test for 150 MCs ...................................... 100

7.5: Relative Frequencies and GPD Probabilities of the General and Cluster 1 Models............................................................................................................. 102

7.6: Relative Frequencies and GPD Probabilities of the General and Cluster 2 Models............................................................................................................. 103

7.7: Relative Frequencies and GPD Probabilities of the General and Cluster 3 Models............................................................................................................. 104
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Network-Side Approach</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Client-Side Approach</td>
<td>4</td>
</tr>
<tr>
<td>3.1</td>
<td>Fréchet, Weibull and Gumbel Distributions</td>
<td>26</td>
</tr>
<tr>
<td>3.2</td>
<td>Generalized Pareto Distribution (GPD)</td>
<td>28</td>
</tr>
<tr>
<td>3.3</td>
<td>Mean Residual Life Plot (MRL)</td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>Parameters Estimation against Thresholds Plot (PET)</td>
<td>30</td>
</tr>
<tr>
<td>3.5</td>
<td>Proposed CBSMA System Architecture</td>
<td>33</td>
</tr>
<tr>
<td>3.6</td>
<td>Mean Excess Plot for Extreme Delay Measurements over Low Bandwidth When the Number of Mobile Clients is 50</td>
<td>41</td>
</tr>
<tr>
<td>3.7</td>
<td>Predicted Extremes for a Different Number of Mobile Clients</td>
<td>44</td>
</tr>
<tr>
<td>3.8</td>
<td>True Positive Ratio for a Different Number of Mobile Clients</td>
<td>45</td>
</tr>
<tr>
<td>4.1</td>
<td>LOP Probabilities for 14 MCs Using Server 1</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>TPR of Trusted MCs vs. All MCs for Servers 1, 4, 6, 8</td>
<td>56</td>
</tr>
<tr>
<td>4.3</td>
<td>Number of Trusted MCs and All MCs Participating in Service Assessment Process for Servers 1, 4, 6, and 8</td>
<td>57</td>
</tr>
<tr>
<td>5.2</td>
<td>SSE Values Produced by the LDDA Algorithm</td>
<td>65</td>
</tr>
<tr>
<td>5.3</td>
<td>GEV probabilities vs. Delay Extremes for Link ( (l_i) )</td>
<td>67</td>
</tr>
<tr>
<td>5.4</td>
<td>True Positive Ratio (TPR) of Extreme Delays of All Routes Used by MC( _5 )</td>
<td>68</td>
</tr>
<tr>
<td>5.5</td>
<td>Prediction Sensitivity of the Delay Extremes of all Routes that Originate from MC( _5 ) (based on the topology described in Figure 5.1.b)</td>
<td>69</td>
</tr>
<tr>
<td>6.1</td>
<td>The Proposed Client-Side Performance Evaluation Architecture</td>
<td>76</td>
</tr>
<tr>
<td>6.2</td>
<td>CloudStack Infrastructure Hierarchy</td>
<td>80</td>
</tr>
</tbody>
</table>
List of Figures—continued

6.3: (6.3.a) MRL Plot of VM$_5$, (6.3.b) PET Plot of VM$_5$ ................................. 81
6.4: Lower and Estimated Bound Values ($P_L$ and $P_E$) for VM$_5$ .................. 82
6.5: Critical Region (CR) and Safe Region (SR) of the Active VMs ............... 84
6.6: GPD Model Improvement with VM Extreme Measurements Exclusion ...... 84
7.1: Illustration of the Clustering Process Used in the ESBCH ....................... 92
7.2: ESBCH Clustering Outcomes for 105 MCs .............................................. 96
7.3: The Goodness of Fit of General, Cluster 1, and Cluster 2 GPD Models .... 97
7.4: The GPD and Calculated Relative Frequencies of the General and Cluster 1 Models ................................................................. 98
7.5: The GPD and Calculated Relative Frequencies of the General and Cluster 2 Models ................................................................. 98
7.6: The Goodness of Fit of the General, Cluster 1, Cluster 2 and Cluster 3 GPD Models ................................................................. 100
7.7: ESBCH Clustering Outcomes for 150 MCs .............................................. 101
7.8: The GPD and Calculated Relative Frequencies of the General and Cluster 1 models ................................................................. 102
7.9: The GPD and Calculated Relative Frequencies of the General and Cluster 2 Models ................................................................. 103
7.10: The GPD and Calculated Relative Frequencies of the General and Cluster 3 Models ............................................................... 104
CHAPTER 1
INTRODUCTION

1.1 Background and General Overview

The vast demand for mobile services is resulting in new challenges for traditional network management. Indeed, network management is a concept that still needs more attention from network's researchers to clarify the network entities' behavior in a steady network environment. As in a network environment, network's entities can be organized in different topologies that required great management effort while providing users with services. Thus, a network management is a set of tools, techniques, and systems that can be employed to aid network specialists’ to monitor, analyze, and manage network entities and discover if they are operational and operating with predefined service limit conditions [1] [2] [3].

International Organization for Standardization (ISO) has defined a model for network management called Fault Management, Configuration Management, Accounting Management, Performance Management, and Security Management (FCAPS) in order to provide a better understanding of the main network management functions. The Fault Management model stands for all operations that deal with detect, log, and report imminent network's failures. Furthermore, the Fault Management model utilizes tools that detect network symptoms and proactively resolve the network problems. The Configuration Management allows monitoring network and system configuration so that the software and hardware information changes can be tracked and managed. In an Accounting Management, network manager ensures to share and control all network resources among a network group or individual, and that, in turn, will limit the network
management issues. Security Management intends to provide two major tasks: better access control to the network resources, and detect and prevent any attacks that cause network failure. Finally, the Performance Management is an attractive substance to most service providers in the sense that it deals with studying network behavior through gathering and analyzing network measurements [2]. This can be done through utilizing different management tools, such as network monitoring tools, to overcome network problems. The performance data that are collected from the monitoring process can help to recognize the network components that are heavily utilized, and, in many scenarios, can find answers to other potential network issues [4]. Furthermore, Performance Management uses techniques that determine network trends and predict the network performance degradations. Consequently, these techniques will help in improving the poor network performance by tuning up the network resources and resolve issues resulted from an overwhelmed network.

Nowadays telecommunication studies are more focused on expanding the research base to cover other important aspects, such as network Quality of Service (QoS), in the network performance management. Researchers have introduced different QoS definitions that fall mostly in the same meaning. A simple network QoS means to deliver network services with predefined requirements that guarantee a certain level of service performance to the network clients [5] [6]. In order to achieve that, a commitment can be established between the service providers and the clients to assure service usage right for both parties. Service Level Agreement (SLA) is a service contract that defines the level of the service provided to the client by the service provider, and the conditions that the client needs to meet in order to receive that QoS stated in the SLA agreement.
Unpredictable service degradations and failures may result in violations of the underlying SLA. Such violations should be eliminated or at least minimized when the client complies with the conditions of the SLA.

Providing an efficient QoS measurement model is a challenging problem in today’s mobile computing and telecommunications networks. Currently, most of QoS techniques utilize service measurements that are collected by the network elements (i.e., network-side monitoring) to evaluate the network’s performance. However, this process does not take into account the service performance from the clients' perspective and might contradict with the SLA. Figure 1.1 illustrates the network-side approach.

![Figure 1.1: Network-Side Approach](image)

In order to overcome the limitations of network-side QoS monitoring, a number of research studies have been conducted to present alternative architectures and algorithms for client-side QoS service assessment in computer networks. The client-side
QoS approach gives the major role to the clients to evaluate the dedicated services through gathering the network measurements and reporting the information. The service providers consider clients’ feedback to revise and resolve service performance issues. This model can be considered as a cooperative model that provides a compromised service plan for both the service providers and the network clients to achieve a better network service performance. Figure 1.2 illustrates the client-side approach.

![Figure 1.2: Client-Side Approach](image)

### 1.2 Problem Statement

Maintaining a reliable QoS model for Mobile Clients (MCs) has to go beyond traditional mobile network management. QoS guarantees offered by Service Providers (SPs) have a meet a minimum level of service performance requirements. However, in most scenarios, service providers' efforts to provide such guarantees can face a challenge
in its continuous service evaluation process while managing SLAs for mobile clients. Mobile service providers are committed to offer high quality services to clients as demanded based on predefined SLA.

As we mentioned previously, measuring network service performance can be accomplished at the service-side or the client-side. The service-side approach utilizes service measurements that are collected by the network elements to evaluate the service performance. On the other hand, the client-side QoS approach is used to monitor and assess the service performance by allowing mobile clients to participate in the evaluation process. In this research, we argue that the clients can provide important input that can be used by the service providers to tune the performance of their service offerings. Since clients play a proactive role in data collection of this model, the service providers can draw an overall view that reflects the services’ behavior from the clients’ point of view. The client-side model provides a major enhancement to the current models that operate in the dark by collecting incomplete data from the service provider’s perspective. This data cannot fully reflect the customer’s experience with the requested services. Furthermore, this model empowers customers by giving them quantitative evidence that could be used to compare the performance of similar services offered by different service providers. Therefore, it is required to exploit another approach that deals mainly with clients' evaluation of the offered services and reflects the service behavior from that view. This approach is called client-side approach.

In this research, we study the tradeoffs between service-side and client-side QoS monitoring and present a client-based architecture for the evaluation and prediction of service degradations in mobile networks. The proposed architecture delegates the role of
QoS measurements collection and reporting (delay and bandwidth extremes) to the MCs. Hence, we consider readings as extremes when they exceed a predefined threshold. MCs and the Broker Manager (BM) are the core parts of the proposed architecture. MCs could be thin clients (users with mobile devices supplied with lightweight monitoring applications), or fat clients (network communication portable equipment that capable to request services and provide rich functions to evaluate the services), whereas the BM is deployed on distributed systems. This architecture deploys efficient techniques, such as Generalized Extreme Value Theorem (GEV), Generalized Pareto Distribution (GPD), Linear Opinion Pool (LOP) and Linear of System Equations (LSE), as a core part of its underlining components. By using these techniques, the proposed architecture guarantees the validity and the accuracy of the performance evaluation process through utilizing robust algorithms that exclude un-trusted monitoring data.

We need to emphasize here that in this research we focus on the collecting the network measurements from the client side regardless of the processing and management locations. Hence, our architecture is scalable enough to be implemented and operated in distributed single and multi-hop QoS network environments such as the Global Environment for Network Innovations (GENI), cloud computing, and Hadoop/MapReduce environments.

1.3 Research Goals

Our objective in this research is to adopt an alternative service monitoring and evaluation approach that addresses the tradeoff between the clients’ demand of the service and the SLA while maintaining a certain level of QoS. Our research is mainly focused on the clients’ experience and their role to evaluate the services offered by
service providers. Consequently, the proposed client-based QoS architecture preserves the overall service performance through predicking the service degradation, and proactively reports the feedback to service provider through the direct management of the service Broker Manager (BM). To accomplish our goal, the following points were addressed:

1. The network measurements (delay and bandwidth extremes) can be collected by MC and under the BM direct supervision and management.

2. Evaluating the efficiency of applying the Extremes Value Theorem (EVT) through fitting the GEV on the collected data. Our proposed Quality of Service Evaluation Algorithm (QoS-EA) algorithm utilizes GEV in conjunction with Joint Probability Distribution (JPD) to assess the service performance and predict potential mobile service degradation over time.

3. Assessing the effectiveness of utilizing LOP in conjunction with GEV in the service evaluation process. The main objective of our proposed algorithms, Behavioral Assessment LOP Algorithm (BALOPA) and Mobile Clients Fidelity Algorithm (MCFA), is to construct a QoS model that excludes the out-of-profile data collected from the MCs, such that any MC with unreliable data is considered as un-trusted and can be excluded from the service evaluation process. Our proposed algorithms are a step forward toward offering the service providers with a better and reliable assessment tool to evaluate and improve their services.

4. Evaluating the service performance through collecting service measurements from MCs in a multi-hop network environment. The proposed algorithms, Link Delay-Driven Algorithm (LDDA) and Link Delay Performance Assessment Algorithm
LDPAA), utilize a System of Linear Equations (SLE) and GEV techniques to predict network link performance degradation by estimating the delay extremes on each hop of a given network topology. Consequently, service performance can be evaluated and improved through a continuous assessment process of the network’s links behavior over time.

5. Verifying the ability of using the client-based QoS architecture for the early detection of cloud service degradations. The proposed algorithm, On-Time Cloud Service Assessment Algorithm (OTCSA), employs the GPD approach to converge to a precise QoS model based on collected delay measurements from the cloud’s mobile clients. Furthermore, by applying a data aggregation process, our approach is capable of providing multi-level service performance assessment through analyzing the collected extreme measurements from Virtual Machines (VMs), zones and datacenters.

6. Evaluating the efficiency of exploiting the strength of the Social Network Analysis (SNA) principles jointly with the GPD to construct consistent QoS models. Our goal is to build QoS models to predict the performance of mobile clients that exhibit similar behavior. Thus, the developed algorithms, Extreme Social Bond Clustering Heuristic (ESBCH) and Immediate Service Performance Assessment Algorithm (iSPA), analyze the strength of the interconnection links between MCs and cluster related MCs in communities of similar behavior.

1.4 Dissertation Outline

This chapter presents the general outline of the proposed client-based QoS monitoring and assessment architecture. The remaining of the dissertation is organized as
follows. Chapter 2 provides summary of the recent literature that addresses the different QoS monitoring and evaluation drawbacks. The chapter also provides in-depth review of the general techniques that undertake the network QoS monitoring issues from the client’s perspective and the solutions that can be applied to eliminate or reduce these issues. Chapter 3 presents the modeling and the theoretical part of the proposed client-based architecture. Through this chapter, we provide a complete illustration of the design of our proposed architecture and its underlying components as well as the advantages of adopting the EVT modeling techniques. Furthermore, as part of the design process of the proposed architecture, we develop new algorithms that exploit GEV and JPD to predict potential mobile service degradations in the client-based QoS environment. Simulation and empirical results are also presented in chapter 3. Chapter 4 introduces new algorithms that utilize GEV and LOP to construct reliable QoS models that can be used in the service evaluation process. Through this chapter, we argue that the network measurements should be collected only from trusted clients while the un-trusted clients have to be excluded in order to create precise QoS models to improve the service evaluation process. Furthermore, chapter 4 shows promising simulation and experimental results after applying our proposed techniques. We present in chapter 5 a novel approach that evaluates service performance through collecting service measurements from MCs in a multi-hop network environment. The proposed approach utilizes a System of Linear Equations (SLE) and GEV techniques to predict network link performance degradation by estimating the delay extremes on each hop of a given network topology. Chapter 6 introduces our adaptive view by applying the suggested client-based QoS architecture in a cloud computing environment. Additionally, this chapter shows the advantages of
utilizing the GPD modeling to predict the VMs behavior and create more efficient services. Moreover, this chapter presents the strength of the presented technique from the simulation and experimental results. In chapter 7, we focus on utilizing SNA principles in our proposed system. According to the effective SNA features, our developed algorithms and techniques were exploited to construct an accurate QoS model which can be used to predict the performance of mobile clients that exhibit similar behavior. We show that the simulation and experimental results demonstrate the benefits of exploiting the SNA to improve the QoS of the offering services. Finally, chapter 8 concludes the dissertation and highlights future research directions.
CHAPTER 2

LITERATURE REVIEW

2.1 General Overview

Maintaining a reliable Quality-of-Service (QoS) model for Mobile Clients (MCs) has to go beyond traditional mobile network management. Thus, service quality assessment has been studied widely in the last years to overcome the QoS issues.

Toward better understanding of the QoS utilization and assessment, QoS models can be classified into three approaches: intrinsic QoS, perceived QoS, and assessed QoS. Intrinsic QoS is the measurement of the network values such as bandwidth, latency, jitter, and delay. The perceived QoS reflects the network’s users’ experiences and assessment of using the provided service. The assessed QoS represents the users’ decision whether they continue to use provided services or not [7].

Back to what we mentioned before, measuring network service performance can be accomplished at the service-side or the client-side. Numerous researchers have intensively investigated the service-side approach to provide better QoS guarantees [8] [9] [10] [11]. Unfortunately, this approach does not take consideration the clients' perspective of the service assessment, if we consider the perceived QoS model, which limits the benefits of the service evaluation process and might not guarantee the Service Level Agreement (SLA).

Lately, researchers and practitioners have studied and evaluated different architectures that utilize the client-side QoS approach to achieve mobile service monitoring. These studies have demonstrated how to efficiently utilize the client-based
QoS approach in a network system to reduce and eliminate network degradations and improve the QoS.

Through this chapter, we review methodologies presented in the literature that adopt the client-based QoS approach partially as a core part of their underlying architecture. Furthermore, this Chapter follows the studies that employ the EVT, LOP, and SNA techniques as effective tools that can be combined with several service performance models to predict the network service degradation.

Wang et al. [12] studied the QoS attributes that can be measured and evaluated from the client-side in order to develop a Web Service selection model. The proposed model built multiple-level cache architecture to improve and speed-up the selection process. Furthermore, the developed architecture has managed to deal with clients and caches similarity such that the selection accuracy of the historical service information can be enhanced. This research has resolved two important issues in the web selection model, the access time of the requested web service and accuracy of the retrieved historical service information.

Li et al. [9], presented an SLA-driven QoS Management Platform which establishes a service level agreement between the service providers and customers. Such a QoS management platform allows clients to specify the QoS requirements, and enables service providers to offer different QoS levels, negotiate with their customers on the possible quality levels and adapt the resource allocation to optimize the system overall performance.

Cardoso et al. [13] proposed a mobile-agent based infrastructure for QoS negotiation, which is part of an entire QoS management architecture. The infrastructure
has a centralized management node that monitors data from networked hosts. The monitoring system allows applications to track service levels and provides hooks to trigger application adaptation. This system is centralized; thus, has a single-point of failure. Even if it replicated, the performance penalty will be significant.

Mahajanet et al. [14] presented a Content-Aware Bandwidth Broker (CABB) that provides adaptive brokering for networked multimedia applications. CABB allocates network resources to multimedia flows based on client requirements, the adaptability of the application, and its tolerance to network level parameters such as bandwidth, delay, and latency. Also, it has been adapted to the network state and reduced QoS rather than completely disrupting application flows.

In [15], Oberortner et al. presented an Architecture Design Decision Model (ADDM) that measures, stores and evaluates performance-related Quality-of-Service agreements in service-based systems. The ADDM is designated to collect decisions patterns in order to discover and prevent SLA violations. However, the proposed model can be resided either on the service provider's or client's network. Adopting a monitoring model to be applied on one of these networks would limit the trust and security, and provide an inadequate service performance assessment.

Thio et al. [16] presented a client-side QoS performance analysis framework for web services (WS) that utilizes two processes that are based on service clients' experiences. The on-going analysis process summarizes and creates WS profiles and client profiles, while the on-demand recommendation process utilizes these profiles to evaluate the WS clients’ experiences. However, the proposed framework does not provide a mechanism to verify or enhance the accuracy of the collected performance data.
Also, another study has been conducted by Thio et al. [17] to develop a client-side QoS performance monitoring and evaluation approach for web service (WS) selection. This study has explored the factors that cause the variation of clients’ performance through analyzing the clients’ historical performance data. Furthermore, the performance assessment process has utilized the client grouping and profiling by examining the clients geographical location as an evaluation factor and how that impacts the prediction of the service behavior.

In [18], Inácio et al. proposed a model that correlates QoS parameters and Quality of Experience (QoE) factors from the user’s perception of the quality. This research intended to combine the QoS measurements with human subjective metrics. The results have illustrated the usefulness of the proposed model when data aggregation and linear regression techniques are applied to the collected data.

Ye et al. [19] proposed a QoS-aware model for Web Service Discovery (WSD). This model allows service providers to dedicate their services without imposing a modification on the standard Universal Description, Discovery and Integration (UDDI) interface and plugs a client-side software in the UDDI server in a transparent way. Clients who participate in the service performance measurement, provide feedback to the QoS Broker about the SLA violators in an interactive manner. Therefore, the QoS Broker arbitrates based on the clients' assessment data, and takes an action to punish the SLA violators.

Lafuente-martinez et al. [20] studied and evaluated the possible servers’ selection strategies that are used to estimate the QoS parameters such as delay, available bandwidth and packet loss. They proposed an approach that allows the client-side server selection
methods for better bandwidth estimation by addressing and overcoming some limitation such as the size, number and condition of the exchanged packets. Also, the proposed approach has utilized a two-step algorithm which estimates the available bandwidth by characterizing the path to each server.

Naomi et al. [21] proposed a client-side service quality approach that utilizes the server log data to evaluate the multimedia streaming application performance. The proposed approach tends to measure the correlation between the end-user service quality and client application through examining the recovered packets from the server’s log data. Hence, this approach gives the service providers an adequate tool to measure the clients’ satisfaction of the offered services. However, this approach does not take the reliability issue into consideration since the log data might be located on a server that might have suffered from single point of failure.

Serhani et al. [22] presented a QoS broker-based monitoring and assessment architecture to support the clients’ demand of the provided web services. The core base of the proposed architecture includes service broker which runs two-phase verification and management techniques. Both phases conduct the verification of the syntactic and semantic of the QoS parameters, and measuring those parameters against their predefined counter parts. However, this approach is cost effective since the broker has to be declared to operate in both sides (server and client sides) before the QoS monitoring and management processes are started.

Kulnarattana et al. [23] presented client-side QoS model for web service selection. The proposed model utilizes uncertainty QoS attributes: response time, availability and reliability as generic criteria for web services selection process. Also, it
adopted the developed Non-parametric test technique which evaluates the historical QoS information collected from client side as well as various network measurements such as bandwidth. The collected information plays significant role in the service selection since it represents the clients’ perspective regarding the change of the services behavior over the use of the dedicated services.

Jin et al. [24] proposed an integrated QoS approach which combines the client-side and server-side models for HTTP traffic analysis in order to preserve the stability of web services. The client-side model utilizes an analysis method for each client’s application that generates HTTP traffic, and classifies and forwards the classified traffic to the server-side. The classified traffic carries out significant information regarding the amount of access that has been made for each site, and provides the service provider with an indication of the reliability of their infrastructure. While the client-side classifies and analyzes the generated traffic, the server-side has the ability to choose the right server location to forward the traffic such that the QoS can be enhanced and avoid SLA infringement.

Petrova-antonova [25] proposed a web service selection QoS-aware approach based on the probability evaluation of the collected data of the QoS properties. The collected data can be extracted from the log files and can be used for prediction of QoS future values. The proposed approach has the benefit of applying the probability mass function (PMF) to the random variable values and calculates the probability of future events that describe the web services. Thus, the web service with maximum probability can be chosen to fulfill the clients’ requirements and be used to estimate the QoS values.
Hence, this approach allows the clients to choose the web services that satisfy their demand and fit with the predefined and expected QoS properties.

Extreme Value Theorem (EVT) has been used as an effective model in the literature to analyze network traffic behavior. EVT provides the ability to model the stochastic behavior of an event at different time scales. The sample maxima of events converge to one of three families of distributions: Fréchet, Weibull, and Gumbel. The GEV provides a generalized distribution that joins and reformulates these distributions into a single family of models [26]. Moreover, the GPD model represents another category of EVT that is mainly used for discovering and modeling the extremes that exceed a predefine threshold. More detailed discussion of EVT is provided in the next chapter.

Dahab et al. [26], presented an EVT based model to predict extreme events and the burstiness of network traffic. Using an EVT model, traffic can naturally be classified into internal and external traffic. These types can be included in service level agreements as traffic descriptor parameters to improve quality of service.

Liu et al. [27], studied the possibility of using EVT to analyze the characteristics of wireless network traffic. In their work, the proposed EVT model is used to estimate the extreme behavior of network traffic based on a pre-define threshold. By fitting the EVT model with the empirical distribution of traffic, the lowest average deviation can be recognized, and can be compared with Exponential, Lognormal, Gamma, and Weibull distributions. The results show that the EVT has a good application potential in the analysis of wireless network traffic.
Our previous study [28] addressed the tradeoffs between network-side and client-side QoS monitoring and presented a client-based architecture for the evaluation and prediction of service degradations in mobile networks. Our proposed architecture delegates the role of QoS measurements collection and reporting (delay and bandwidth extremes) to the MCs. Hence, we consider readings as extremes when they exceed a predefined threshold. The reported data are provided to the Broker Manager (BM) which utilizes GEV to predict potential service degradations and provide service providers with global information about the QoS level throughout the mobile network. The GEV based model underlies the strength of our approach as it helps to recognize and model the service performance fluctuations throughout the mobile network.

Linear Opinion Pool (LOP) [29] [30] [31] is a technique for combining individual probability distributions using a weighted average to calculate an unknown variable $\theta$. LOP can serve as an effective tool for service consumers and providers to obtain accurate observations that summarize the service behavior over time while minimizing the chances to incorporate noisy and misleading observations reported by the clients. More detailed discussion of LOP is presented in the next chapter.

In [32], Buchegger et al. presented a Bayesian approach for a reputation system that aims to isolate misbehaving nodes in mobile ad-hoc networks. The proposed approach merges multiple probabilistic models using a Linear Opinion Pool (LOP) based-scheme. This scheme is used to detect nodes that are reporting observations that deviate from the trusted median of observations.

In our recent paper [33], we proposed a client-based QoS service measurement model that utilizes LOP to build up a precise QoS model based on trusted observations.
while excluding noisy and misleading observations reported by untrusted MCs. Our proposed approach serves as a reputation system that ensures the reported MC data meet a certain level of trustworthiness in order for it to be included in the service evaluation process. Furthermore, our model is versatile and can be employed in mobile network irrespective of the underlying technologies (e.g., 3G, 4G, LTE, WiMax, etc.).

Many research studies have focused on single-hop networks to monitor and evaluate the network service performance. However, this approach has some limitations since it is not reliable enough to reflect the network behavior in the monitoring process. Instead, utilizing multi-hop networks provides more efficient way to efficiently model and predict the network performance.

In [34] Lahyani et al. utilized the EVT to analyze QoS measurements through developing a monitoring module that detects QoS degradations in publish/subscribe (P/S) systems. The proposed approach aims at discovering immediate link failures between brokers by utilizing the Gumbel and Gaussian distributions. However, this approach aims at building a monitoring module in the network elements instead of application, and that would limit the mobility of the monitoring process.

Gorbil et al. [35] proposed a network-layer solution that supports multiple QoS traffic type for multi-hop wireless mobile ad-hoc networks. The presented solution combines link-state topology with on-demand cost dissemination in a form of hybrid routing protocol. Also, it adopts QoS path selection under dynamic network conditions. However, we can still classify this solution as a service-side QoS approach.

We presented in [36] a client-side novel approach that evaluates service performance through collecting service measurements from mobile clients (MCs) in a
multi-hop network environment. The proposed approach utilizes SLE and GEV techniques to predict network link performance degradation by estimating the delay extremes on each hop of a given network topology. Consequently, service performance can be evaluated and improved through a continuous assessment process of the network’s links behavior over time.

2.2 QoS and Cloud Computing

Cloud computing is evolving to become a big leap in the computing revolution. Cloud computing can be defined in many different ways, but the most realistic cloud computing definition that is related to our study is a vast resource pool that can be used to provide resources in a timely manner upon customers’ demand and under customers’ satisfaction. Several researchers have started to study network issues that are stemming from the interaction between cloud clients and cloud service providers. Quality of Service (QoS) is one of the challenging issues that can impact the cloud performance through lack of resource provisioning, scalability, and poor management.

Katsaros et al. [37], presented a multi-layered service monitoring system that collects and aggregates measurements from Cloud-based Systems. The operation of the proposed system, besides collecting measurements, is designed to provide on-the-fly system reconfiguration that permits system self-adaptation in real-time; thus, meeting QoS guarantees and improving the system’s interactivity.

Junwei et al. [38], proposed a resource monitoring model for cloud computing platforms. The proposed model combines Virtual Machine Monitor (VMM) and a set of C/C++ and Java libraries to collect real-time static and dynamic information for of the
cloud’s VMs. However, this model suggests building the resources’ data collectors and the monitoring objects at the nodes that provide the cloud’s services.

Another study has been conducted by Chauhan et al. [39] to perform a measurement-based analysis of the impact of running Video-On-Demand (VOD) servers in parallel in a virtualized environment. The study focused on the QoS measurements collected by clients to evaluate the VMs that suffer service degradation in presence of overwhelming VOD server requests. Also, this study argued that the client side’s view is important in the sense that the misbehaved servers will impact the reliability of the service delivered to the clients.

Our previous work [40] has investigated the possibility of applying the client-based approach for measuring the service performance in the cloud computing environment. We presented a technique for early detection of cloud service degradations through utilizing GPD. The proposed technique employs the GPD to construct an accurate QoS model by fitting the network measurement (extremes) collected by the reserved Virtual Machines (VMs). Furthermore, by applying a data aggregation process, the proposed approach is capable of providing multi-level service performance assessment through analyzing the collected extreme measurements from VMs, zones and datacenters.

Emeakaroha et al. [41] developed the Detecting SLA Violation infrastructure (DeSVi) for early discovering of SLA violation in the cloud environment. The dedicated cloud resources can be monitored using special framework (LoM2HiS) which maps low-level resource metrics to user defined SLA. In order to detect the SLA violations, the proposed infrastructure predefines and exploits service level objectives and construct
knowledge database such that violated service request that violates SLA can be prevented. Correspondingly, the presented approach has offered a service cost management technique by which unnecessary penalties can be avoided, and improved the offered services through proactively responded to SLA violation generated by clients in a timely manner.

2.3 QoS and Social Network Analysis

Adopting new research ideas in network behavioral analysis have been the focus of diverse studies that deal with critical issues in the networked systems field. These studies investigate the possibility to undertake a new dimension of network behavioral analysis by means of utilizing Social Network Analysis (SNA) techniques to study the network as an active community and distinguish the structure and the characteristics among the network’s nodes. SNA has added a step forward towards differentiating the network topologies through exploring the strength of the network’s underlying connections [42]. In the next chapter, a thorough review of SNA principles and techniques will be provided.

In the literature, several researchers have been conducting studies to investigate the network’s structure and analyze the network communities’ behavior by employing SNA techniques.

Barzinpour et al. [43], proposed an algorithm to compute closeness centrality and detect communities in complex networks. The proposed algorithm aimed to partition a complex network through adding individual node attributes into a D-dimensional space using eigenvectors of the Laplacian matrix. The Euclidean distance and k-means clustering techniques are then used in the eigenvectors space to generate network’s
partitions. However, determining the number of communities is a prerequisite stage that should be performed before generating network partitions, and that can be done through using spectral clustering. The results have shown that the proposed algorithm has detected inter-cluster closeness centrality and the number of clusters.

Another study has been conducted by Li et al. [44] to capture the topological and semantic of heterogeneous social networks. Their effort has resulted in a knowledge discovery framework that has three models, tensor-based relational adjacency model, contribution-based, diversity-based and similarity-based centrality, and role-based clustering schema. The role-based clustering scheme has adopted the social positions instead of the community structure clustering. This scheme clusters network nodes depending on their higher-order relational connection in the network.

Xu et al. [45], presented a structural clustering algorithm for networks (SCAN) to identify clusters, hubs, and outliers in networks. The proposed algorithm has employed vertices neighborhood as clustering measure instead of using the direct connection among nodes. The nodes that share more neighbors can be grouped as a cluster of the same community.

Meng et al. [46] proposed a spectral clustering approach which analyzes the network structure and determines the number of clusters using social network analysis principals. This approach aims to overcome the limitations of the traditional clustering techniques since it does not require specifying the number of the required clusters in advance. The presented approach utilized the degree-centrality and betweenness-centrality for ranking the network nodes, and categorizing them as leaders and members. Hence, the nodes can be recognized as leaders among other when they demonstrate close
relationship with others and maintain higher centrality. Therefore, the number of created network clusters can be obtained from the number of classified nodes’ leaders.

Our previous work [47] has utilized client-based QoS as a new and alternative approach to assess the performance of networked and cloud-based services. It is a step forward towards predicting service failures and degradations. We exploit the SNA techniques in conjunction with GPD to evaluate the service performance from clients’ point of view. Correspondingly, we presented a novel clustering heuristic (ESBCH) that uses the SNA principles to group the MCs into communities based on their relationships. The proposed heuristic proactively identifies clients that exhibit similar behavior through the Kendall-Tau statistic. Furthermore, we utilized GPD models as an effective prediction tool to measure and evaluate the performance of clustered MCs.
CHAPTER 3
CLIENT-BASED QOS SYSTEM DESIGN AND MODELING

3.1. Introduction

The effectiveness of a QoS model in a network environment should fulfill both network clients’ and service providers’ expectations. This assumption is not easy to achieve since the behavior of the services offering cannot be predicted while network’s components undergo performance fluctuations. Therefore, our vision is that adopting client-based QoS approach is a step forward to provide an efficient way to maintain stable and reliable service that, in turn, impacts the network QoS. Hence, this can be done by predicting the service degradation and proactively reporting adequate feedback to service providers so that the service offering can be improved efficiently.

In this chapter, an intensive and detailed illustration of the proposed client-based QoS system is provided. Furthermore, in the following sections, we present several terms and techniques that represent the core part of the proposed architecture.

3.2. Extreme Value Theorem (EVT)

Extreme readings are in general represented in the tail of a probability distribution function. Extreme Value Theory (EVT) is a powerful statistical tool for modeling the tail of a statistical distribution [48] [49] [50] [51] [52]. It provides the ability to model the stochastic behavior of an event at different time scales and determines the trend in data. Moreover, EVT, of both type GEV and GPD, has widely been used to overcome the limitation of Central Limit Theorem (CLT). Generalized Pareto Distribution (GPD) and Generalized Extremes Value Distribution (GEV) can be categorized as EVT.
3.2.1. GEV

The GEV utilizes Block Maxima (BM) technique to divide the collected data “into non-overlapping blocks of the same length and choosing the maximum from each block” [51]. The GEV distribution joins and reformulates three distributions, namely: Fréchet, Weibull, and Gumbel into a single parameterized distribution that can model the BM’s extreme.

\[
\begin{align*}
\text{Fréchet} : \Phi_\alpha(x) &= \begin{cases} 
0, & x \leq 0 \\
\exp(-x^{-\alpha}), & x > 0, \quad \alpha > 0,
\end{cases} \\
\text{Weibull} : \Psi_\alpha(x) &= \begin{cases} 
\exp(-(-x)^\alpha), & x \leq 0 \\
1, & x > 0, \quad \alpha > 0,
\end{cases} \\
\text{Gumbel} : \Lambda(x) &= \exp(-e^{-x}), x \in \mathbb{R}.
\end{align*}
\]

Fig. 1. shows the PDF shapes for the standard Fréchet, Weibull and Gumbel distributions

![Figure 3.1: Fréchet, Weibull and Gumbel Distributions](image)

The GEV model is given by [53]:
Where \( x \) is an extreme such that \( 1 + \xi x > 0 \), and \( \xi \) is a shape parameter that can take the value of \( \xi = \alpha I \) for the Fréchet distribution, \( \xi = -\alpha I \) for Weibull distribution and \( \xi = 0 \) for Gumbel distribution.

### 3.2.2. GPD

The GPD provides a useful tool to evaluate and model short-term extreme observations (e.g., hourly and daily extreme events). Moreover, it utilizes threshold excess techniques to quantify the extremes that exceed predefined thresholds. The threshold selection techniques are called Peak over Threshold (POT). Alternatively, the Generalized Extreme Value (GEV) models the long-term extreme observations using the Block Maxima (BM) technique. Adopting the GPD approach has been widely documented in the literature to model network traffic [54]. The GPD is given by [55]:

\[
H_\xi(x) = \begin{cases} 
  e^{-(1+\xi)^{-1/\xi}}, & \text{if } \xi \neq 0 \\
  e^{-e^{-x}}, & \text{if } \xi = 0 
\end{cases} \tag{3.2}
\]

Where \( x \) is an extreme value, and \( \xi \) is the shape parameter that can take \( \xi > 0 \) for the ordinary Pareto distribution, \( \xi < 0 \) for Pareto II-type distribution, and it is known as an exponential distribution when \( \xi = 0 \). It is important to mention that the shape parameter is invariant to data block size when compared to that in GEV. That is, choosing a large block size would affect the GEV’s parameters’ values, but not the
GPD’s parameters, especially the shape parameter \([56] [55]\). Fig. 3.2. shows the GPD family.

For the sake of choosing an appropriate GPD model that fits the collected extremes, a suitable threshold should be carefully selected. However, selecting the threshold to be too high would generate few samples and that would lead to high variance whereas selecting the threshold to be too low would lead to bias the distribution model. The objective of threshold selection is to pick a threshold that makes the model provide reasonable approximation of the distribution of the extreme event. Several techniques can
be used to find an appropriate threshold. The Mean Residual Life plot (also referred to as Mean Excess Plot) applies the GPD to a range of thresholds in order to evaluate the stability of the parameters and to determine a good threshold selection [56].

3.2.2.1 Mean Residual Life Plot (MRL): Let’s assume that $x_1,\ldots,x_n$ is sequence of collected measurements, and $x_{(1)},\ldots,x_{(k)}$ is a subset of data points (Extreme events) that exceed a certain threshold, $u$, where \( \{ x_{(i)} : x_{(i)} > u \} \). Define threshold excesses by: $y_{(j)} = x_{(i)} - u$ for $j=1,\ldots,k$. The following set of points define the Mean Residual Life Plot (MRL):

$$\left\{ \left( u, \frac{1}{n_u} \sum_{i=1}^{n_u} (x_{(i)} - u) \right) : u < x_{max} \right\}$$

Where, $x_{max}$ is the largest data set of extreme measurements. The Mean Residual Life Plot should exhibit some linearity above the selected threshold $u$ to consider $u$ as a suitable threshold [56].

3.2.2.2 Parameters Estimation against Thresholds (PET): The idea behind this approach is to fit the GPD model to data each time using a range of different thresholds. The GPD’s shape and scale parameters can be extracted, and the stability of these parameters is then checked. Consequently, the best threshold can be used when it satisfies the stability between the shape and the scale parameters. However, it is recommended to adopt the lowest threshold possible to make the GPD model provide a reasonable approximation of the distribution of the underlying extreme event [56].

Figures 3.3 and 3.4 illustrate an example of selecting the best threshold $u$ after applying PET and MRL, respectively.
Figure 3.3: Mean Residual Life Plot (MRL)

Figure 3.4: Parameters Estimation against Thresholds Plot (PET)
3.3. Client-Based QoS Service Monitoring and Assessment System

While traditional service management architectures delegate the monitoring role to the managed services, our proposed client-based Service Monitoring Architecture (CBSMA) delegates the monitoring role to the clients. This essentially relieves the managed services from all the tasks associated with service monitoring and performance tuning. Furthermore, Service Providers (SPs) rely on the Mobile Clients (MCs) to gather and share end-to-end performance with service providers. In this work, we assume that the MCs and the servers communicate through a single hop wireless channel. The core of the proposed system is the QoS Broker which collects the data from the mobile clients and utilizes EVT to build a model based on the collected extreme measurements. The QoS Broker is scalable and can be implemented on different distributed environments.

Following are the details of the different modules of the proposed architecture:

1. **Mobile Clients (MC):** Mobile clients seek to use available services provided by different service providers. The MCs gather and aggregate parameterized data that pertain to the monitored services. Collected data is then submitted to the broker manager module for further analysis.

2. **QoS Broker:** The role of the QoS broker is to collect data from the MCs and disseminate it to the SPs. This module also provides the clients with thresholds that force them to report QoS measurements when these thresholds are exceeded. The QoS broker can control the values of these thresholds to strike an intelligent balance between the amount of the data collected from the clients and the accuracy of the EVT model. The QoS Broker is comprised of the following components:
• Broker Manager (BM) Module: this module interacts with MCs to analyze the collected data. This module is also responsible for handling SP queries that seek to predict the behavior of the services over time.

• Historical Service Measurements Database: this database archives all service measurements submitted to the QoS broker over time.

• Service Measurement and Modeling (SMM) Database: this database hosts the service measurement and modeling tables. The service measurement table is used to track information about the MCs including delay, bandwidth, number of clients and physical distance from the various SPs.

• EVT Model Builder: this module builder utilizes GEV distribution (Fréchet, Weibull, and Gumbel) and GPD distribution models to fit the collected extreme measurements stored in the SMM database and tunes the models’ parameters according the collected data.

• Database Manager: this module is responsible for updating the SMM database. Furthermore, it categorizes the parameters of the collected data into different levels (e.g., low, medium and high).

3. Service Providers (SPs): Service providers offer dedicated services to the MCs through negotiated SLAs and provide certain levels of QoS. Furthermore, SPs utilize the QoS broker to find efficient and cost effective alternatives to enhance their current service offerings. Also, SPs can subscribe to QoS broker services that enable them to extract information about the QoS perceived by MCs of other competitor SPs.

Fig. 3.5. depicts the main components that comprise the proposed CBSMA system.
The broker manager module assigns monitoring threshold \( \varepsilon \) that is used to collect the performance data from the various clients. For simulation purposes, the end-to-end delay threshold \( \varepsilon \) can be computed as follows:

\[
\varepsilon = \frac{L_{\text{client} \rightarrow \text{server}}}{3 \times 10^8} + \frac{1}{\mu - \lambda} \tag{3.5}
\]

Where \( L_{\text{client} \rightarrow \text{server}} \) is the distance between the client and the server, \( \lambda \) is the service offered load and \( \mu \) is the service completion rate. For the purpose of simulation, we assumed that the queuing delay is based on an \( M/M/1 \) model. Other models can be used in the broker to set this threshold.

In real time scenarios, the threshold can also be assigned using pre-defined conditions that is concluded based on the collected network measurements.
The threshold $\varepsilon$ assigned to the MCs by the BM is used to gather the maxima of delays measured while utilizing services offered by various SPs whenever the client measurements exceed $\varepsilon$, the client sends a data tuple to the QoS broker as follows:

$$<C_{ID}, D, L, B, S_{ID}, T>$$

Where, $C_{ID}$ is the Client ID, $D$ is the delay measurement (extreme value), $L$ is the distance between the client and the server, $B$ is the channel bandwidth, $S_{ID}$ is an ID that indicates the service number and $T$ is the client’s timestamp that indicates the time when the tuple is sent.

The broker manager module forwards the collected tuples to the historical service measurement and modeling database. Upon receiving a request from a given service provider to conduct service evaluation, the broker manager module interacts with the EVT model builder and the database manager modules to retrieve the collected extreme measurements, fit it to an EVT model (GEV or GPD models) and select the best parameters that describe the model.

The broker can choose between Bayesian classifier and Joint Probability Distributions (JPD) to calculate the set of probabilities requested by the service provider based on the fitted EVT models. If the service provider’s request has multiple parameters to evaluate, the broker selects Bayesian classifier in conjunction with EVT model. Such a query is called a Bounded Query. Otherwise, the broker chooses JPD in conjunction with EVT and the query is called Unbounded Query. The probability with maximum value represents the best prediction of service performance of a given tuple.

The service evaluation process provides feedback to the service provider that shows the current and predicted performance of the service. The service provider can
then react to the broker evolution by tuning the service performance. This tuning can be achieved by changing the number of replications of the service, the physical location of the service or by adjusting the amount of resources dedicated for the service offering.

3.4. Verifying The Efficiency of The proposed CBSMA System Architecture

We have conducted several experiments that verify the efficiency of our proposed client-based Service Monitoring Architecture (CBSMA). The proposed architecture [28] has strengthened through developing new approaches and algorithms that are used to resolve client-side QoS network performance drawbacks. The following sections illustrate the proposed algorithms and their outcomes:

3.4.1. Quality of Service Evaluation Algorithm (QoS-EA)

Our proposed architecture is based on Quality of Service Evaluation Algorithm (QoS-EA). The QoS-EA algorithm takes the SMM table and the query (which could be bounded or unbounded) that is received from the service provider as input to build up a model that best fits the aggregated measurements. The QoS-EA algorithm utilizes a combination of extreme value theorem and Bayesian rule classifier to provide a response for the received query. The following pseudo-code describes the QoS-EA algorithm:

**Quality of Service Evaluation Algorithm (QoS-EA)**

**Input:** SMM = Service Measurement and Modeling Table

\[ Q = \text{Service Evaluation Query, and } \{q_1, q_2, \ldots, q_x\} \in Q \text{ where } q_i \text{ is the } ith \]

parameter in query \( Q \) and \( 1 \leq x \leq |SMMparameters| \).

\( Cl = \text{Class category of a selected parameter from Bandwidth (B), Physical Distance (L), and Number of Clients (N)}. \)

**Output:** Predicted probability of query \( Q \)
**Step 1:** Let \( B, L, N \) be class categories from the SMM table,

Let \( B, L, N \in [L_w, M_d, H_g] \): where \( L_w, M_d \) and \( H_g \) are the Low, Medium and High alues, respectively.

Let \( S = [] \): array holds part of SMM table

Let \( u = 0 \): EVT model counter

**Step 2:** Determine the number of clients based on clients’ time variation, and update the SMM table

For  \( \forall \ ci \in C_{id} \)

\[ \text{SMM}[ci, N] = \text{findTimeStamp(), function to compute the number of} \]

clients that sent data to the Broker at time \( t \)

End for

**Step 3:** Choose classification criterion variable \( V \) s.t. \( V \in [B, L, N] \), where \( V_i \) = class category of \( [B, L, N] \), \( 1 \leq i \leq 3 \), where \( i \) is the number of class categories of each parameter (i.e., low, medium and high).

For \( \forall \ V \in [B, L, N] \) Do

For \( \forall \ V_i \in [L_w, M_d, H_g] \) Do

For \( k = 1 \) to \( |\text{SMM}| \) Do

\[ S[k] = \text{SMM}[d_k, V_i] \]

End for

Apply EVT to \( S \)

\[ S_{\text{EVT}<\text{model, params}>}^u = \text{EVT}(S) \]

Increment \( u \) by 1

End for
End for

**Step 4:**  if $|Q| \geq 2$ then

Calculate the probability of $Q$ using Bayesian classifier

Let $V = Cl$

For $\forall V_i \in [L_w, M_d, H_g]$ Do

$$P_{vi} = S_{EVT<model, params,q1>}^{vi} \prod_{k=2}^{|Q|} \frac{|V_{i,q,k}|}{|V_{i,SM}|}$$

$$P_{vi} = P_{vi} \times \frac{|V_{i,SM}|}{|SM|}$$

End for

$$P(Q|Cl) = \text{Max}(P_{vi})$$

Else

Let $z = 0$, and $d = q_1$

Let $MAT_d = \text{Table holds none-zero probabilities of } d$

Calculate the probability of $d$ for all $L$, $B$, and $N$

For $\forall L_i \in [L_w, M_d, H_g]$ Do

$$l_i = S_{EVT<model, params,d>}^{Li} \times P(L_i)$$

For $\forall B_i \in [L_w, M_d, H_g]$ Do

$$b_i = S_{EVT<model, params,d>}^{Bi} \times P(B_i)$$

For $\forall N_i \in [L_w, M_d, H_g]$ Do

$$n_i = S_{EVT<model, params,d>}^{Ni} \times P(N_i)$$

$$d_{prob} = l_i \times b_i \times n_i$$

if $d_{prob} > 0$ then

Increment $z$ by $1$

$$MAT_d[z] = <d_{prob}, L_i, B_i, N_i>$$
End if

End for

End for

End for

$MAT_d = \text{sort}(MAT_d)$

$R = \text{min}(MAT_d[1])$

$P(Q) = R$, and $d \in \text{to class } [R_B, R_D, R_N]$

End if

**Step 5:** End of Algorithm

The QoS-EA has mainly three stages:

1. The first stage categorizes the SMM data into three classes (High, Medium, and Low). This can be done through specifying pre-defined classification conditions for each network measurement. The reason to categorize the SMM data is to provide a certain degree of the evaluation freedom of the service performance and to focus on the overall picture of the service offering behavior. Table 3.a and 3.b show a snapshot of the service measurement and modeling table (SMM) in normal and categorized forms, respectively.

2. The second stage fits a set of GEV distribution models to the SMM data and estimates the distribution parameters. The algorithm applies the Mean Excess Function (MEF) to verify the goodness of fit of the selected model. MEF is defined as follows:

$$e_n(u) = \frac{\sum_{i=1}^{n}(X_i - u)}{\sum_{i=1}^{n}I(x_i>u)} \quad (3.6)$$
Where \( I = 1 \) if \( X_i > u \) and 0 otherwise. The MEF is the sum of the excesses of the data obtained from the GEV model \( (X_i) \) over the threshold \( u \) divided by the number of data points that exceed the threshold \( u \) [55].

**Table 3.1: (a) SMM in a Normal Form and (b) SMM in a Categorized Form**

(a)

<table>
<thead>
<tr>
<th>Client ID</th>
<th>Total Delay ( D ) (s)</th>
<th>Distance ( L ) (m)</th>
<th>Bandwidth ( B ) (kbps)</th>
<th>No. of Clients ( N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0051</td>
<td>783</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>0.1016</td>
<td>314</td>
<td>1000</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>0.0042</td>
<td>582</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>0.0041</td>
<td>587</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>0.0041</td>
<td>561</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>0.0048</td>
<td>715</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>0.0052</td>
<td>783</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>0.1016</td>
<td>314</td>
<td>1000</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>0.1031</td>
<td>582</td>
<td>1000</td>
<td>3</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th>Client ID</th>
<th>Total Delay ( D ) (s)</th>
<th>Distance ( L ) (m)</th>
<th>Bandwidth ( B ) (kbps)</th>
<th>No. of Clients ( N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>L</td>
<td>M</td>
<td>L</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>L</td>
<td>M</td>
<td>M</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>M</td>
<td>H</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>M</td>
<td>H</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>H</td>
<td>M</td>
<td>H</td>
<td>3</td>
</tr>
</tbody>
</table>

3. The third stage applies either Bayesian classifier or Joint Probability Distributions (JPD) techniques to calculate the set of probabilities requested by the service provider.
based on the fitted GEV models. As mentioned before, the technique’s selection process is based on the type of the query request that is required by service providers. If the service provider’s request has multiple parameters to evaluate, the algorithm selects Bayesian classifier in conjunction with GEV. Such a query is called a Bounded Query. Otherwise, the QoS-EA chooses JPD in conjunction with GEV, and the query is called Unbounded Query.

3.5 Simulation Results

To verify the efficiency of the proposed CBSMA and QoS-EA heuristics, a MATLAB-based simulator has been developed to simulate QoS parameters for a set of mobile clients that are accessing a certain service. The simulated QoS parameters include delay and bandwidth under various traffic load conditions from a set of mobile nodes that are exhibiting a Random Waypoint Mobility model [57]. Our simulator (which utilizes the Evim [55] and EasyFit tools [58]) was run on scenarios that included 25, 50, and 100 mobile clients and 8 servers. The MCs move with different velocities in an area of 3000 × 3000 m. The following results have been extracted from the simulator.

To check the degree of fitness of a selected GEV model of the simulated SMM data, the mean excess plot used (Fig. 3.6) shows the collected extreme delay measurements when the bandwidth B = (L) and the number of mobile clients is 50.

From Fig. 3.6, the mean excess plot shows that the collected extreme delay under low bandwidth condition follows a distribution with a tail that is lighter than that of a Gumbel distribution. Since this excess mean plot is almost linear with a negative slope, the behavior can be modeled by using a model with a negative shape parameter.
We conducted two more simulation experiments with the number of mobile clients 25 and 100. Our results illustrate that when we vary the number of mobile clients, the GEV model fits the SMM data to a model with negative shape parameters. Hence, we can generalize the GEV model to the extreme delay data against all bandwidth conditions with any number of mobile clients. Table 3.2 shows the fitted GEV model parameters when the number of clients is 25, 50, and 100.

Figure 3.6: Mean Excess Plot for Extreme Delay Measurements over Low Bandwidth When the Number of Mobile Clients is 50
Another simulation experiment that we conducted involved the submission of the bounded query $P(D = 0.6729| L=H, N=L)$ to the QoS broker manager to predict the bandwidth condition that satisfies the aforementioned query. Using both GEV and Normal distributions, the Bayesian classifier method predicted that query is satisfied under low bandwidth ($B=L$) with the following probabilities:

GEV distribution:

$$P(D = 0.6729| B = L, L=H, N=L) = 0.00375$$

Normal distribution:

$$P(D = 0.6729| B = L, L=H, N=L) = 0.000399$$

<table>
<thead>
<tr>
<th>Number of Mobile Clients</th>
<th>Block Size</th>
<th>Bandwidth</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>High</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>GEV</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$k$</td>
<td>$\sigma$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.1854</td>
<td>1.2164E-6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.2160</td>
<td>1.2457E-6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.0151</td>
<td>8.1778E-7</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>High</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GEV</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$k$</td>
<td>$\sigma$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.1917</td>
<td>8.0264E-7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.1960</td>
<td>8.0675E-7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.0211</td>
<td>4.2257E-7</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>High</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GEV</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$k$</td>
<td>$\sigma$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.1813</td>
<td>8.9930E-7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.1822</td>
<td>9.0932E-7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.0624</td>
<td>4.2015E-7</td>
</tr>
</tbody>
</table>
Our simulation results demonstrate that the Bayesian classifier provides better predictions of the network conditions when it relies on the GEV model as opposed to the Normal distribution.

3.6. Experimental Results and Analysis

To validate our experimental results, we conducted an empirical study in which we collected performance data for a client located in Tampa, Florida through the online Pingdom service. Pingdom is a monitoring tool that can be used to collect network measurements (uptime and downtime) in order to evaluate the website or service performance [59]. In our experimental study, the client (user or machine that located on the Pingdom on-site service) requested a service from Western Michigan University’s library (URL: http://www.wmich.edu/library/) once every hour for nine days. By applying the Normal and GEV distributions to the query \( P(D = 0.917 \mid B = M) \), following probabilities were calculated:

GEV distribution:

\[
P(D = 0.917 \mid B = M) \times P(B = M) = 0.4162
\]

Normal distribution:

\[
P(D = 0.917 \mid B = M) \times P(B = M) = 0.1712
\]

From both calculated probabilities, the Bayesian classifier with GEV predicted \( d \) to belong to the category \((B=M)\) with a probability higher than predicted using the Bayesian classifier with the Normal distribution.

These experimental results concur with our simulation results and illustrate the enhancements that the extreme value theorem can be offered when it is used to predict network conditions.
Beyond the bounded queries discussed in the previous paragraphs, service providers can send unbounded queries to the QoS broker manager. Unbounded queries refer to those that contain only a delay parameter. Other parameters, such as (L, B, and N) are predicted based on SMM data. Fig. 4 illustrates that the number of delay extremes predicted by the GEV distribution is greater than those predicted by the Normal distribution as the number of mobile clients increases from 25 to 50 and then to 100.

![Figure 3.7: Predicted Extremes for a Different Number of Mobile Clients](image)

**Figure 3.7: Predicted Extremes for a Different Number of Mobile Clients**

To evaluate the effectiveness of our approach in predicting various network conditions, true positive and false negative measurements have been used as follows [60]:

- **True Positive (TP):** An extreme delay measurement, with certain parameters (L, B, and N), has been predicted by the QoS broker manager and that matches the collected SMM data.
• False Negative (FN): An extreme delay measurement, with certain parameters (L, B, and N), has not been predicted and that does not match with the collected SMM data.

The QoS broker manager calculates the possible probability of a certain extreme delay value by applying JPD with GEV distributions and matches the extracted probability with the SMM data.

Based on the values of the TP and FN counters, the QoS broker manager calculates the True Positive Ratio (TPR) factor as illustrated in the following equation [61]:

$$TPR = \frac{TP}{TP + FN}$$  \hspace{1cm} (3.7)

Fig. 3.8 shows the TPR value for 25, 50, and 100 mobile clients. The overall TPR maintains a value exceeding 91% irrespective of the number of mobile clients.

![Figure 3.8: True Positive Ratio for a Different Number of Mobile Clients](image-url)
In this stage of our research, we presented a new mobile service management architecture that solely relies on data collected from the mobile clients without the need to insert measurement probes in the core transport network. The proposed architecture is based on an algorithm that exploits Generalized Extreme Value models (GEV) and Joint Probability Distributions (JPD) to predict potential mobile service degradations.
CHAPTER 4
A CLIENT-BASED QOS DATA APPROACH USING GENERALIZED EXTREME VALUE THEOREM AND LINEAR OPINION POOL

4.1. Introduction

In our previous study [28], we assumed that the collected performance data has the same level of accuracy in the service evaluation process. We assumed that all MCs are trusted to report correct data about the performance of the services that they access. However, relying exclusively on this assumption may limit the accuracy of the service performance model as the MCs cannot be trusted to report accurate performance data all the time. Alternatively, our target is to improve the accuracy of the service evaluation process by providing trusted and unbiased data to the service providers in order to enable them build an accurate service performance model.

In this chapter, we propose a client-based QoS service measurement model that utilizes LOP to build a precise QoS model based on trusted observations while excluding noisy and misleading observations reported by un-trusted MCs. Our proposed approach serves as a reputation system that ensures that the reported MC data meet a certain level of trustworthiness in order for it to be included in the service evaluation process. Furthermore, our model is versatile and can be employed in mobile network irrespective of the underlying technologies (e.g., 3G, 4G, LTE, WiMax, etc.).

4.2. Linear Opinion Pool (LOP)

Linear Opinion Pool (LOP) [29] is a technique for combining individual probability distributions using a weighted average to calculate an unknown variable $\theta$. The LOP model is given by:
Where \( w_i \) is a weight such that \( w_i \geq 0 \) and \( \sum_{i=1}^{n} w_i = 1 \), \( n \) represents the number of probability distribution models, \( p_i(\theta) \) is the distribution model \( i \) for unknown \( \theta \). LOP can serve as an effective tool for service consumers and providers to obtain accurate observations that summarize the service behavior over time while minimizing the chances to incorporate noisy and misleading observations reported by the clients.

### 4.3. Proposed Approach

In this model, we adopt a multi-level performance analysis technique to exclude out-of-profile data. Our proposed technique digs deep to discover and evaluate the MCs’ behavior through a statistical analysis of the MCs' history. Hence, our proposed technique provides a mechanism to isolate misbehaving MCs that report unreliable extremes about the service performance. The following steps describe our proposed technique:

**STEP 1:**

Every server is assigned a weight that represents server's performance status. Thus, the server’s weight is proportional to the probability that the server suffers performance degradation. However, to determine how to assign the weights to the individual servers is a subjective matter [62]. In our approach, we calculated the weights by taking the average number of reported extreme measurements reported by the MCs (e.g., encountering high delay and when the bandwidth is high) while using the services of a particular server. Equation (4.1) shows the formula that we have used to calculate the status weight of the individual servers when the MCs reported only high delay extremes:

\[
p(\theta) = \sum_{i=1}^{n} w_i p_i(\theta)
\]

(4.1)
\[ W_i = \frac{|c_j = C_H|}{|c_j = C_H| + |c_j = C_M| + |c_j = C_L|} \quad (4.2) \]

\[ j = |c \epsilon s_i|, 1 \leq i \leq |S| \]

Where \( S = \{s_1, s_2, s_3, \ldots, s_k\}, \quad 1 \leq k \leq |Server|, c_j = \) all MCs' extremes that are associated with \( s_i, \) \( C_H = \) number of MCs that reported extreme performance measurements with high delay when the bandwidth is high, \( C_M = \) number of MCs that reported extreme performance measurements with medium delay when the bandwidth is medium, and \( C_L = \) number of MCs that reported extreme performance measurements with low delay when the bandwidth is low.

After we calculate the status weights for all servers, all collected performance measurement data is organized in a table called the Server Performance Table (SPT). This table also holds the estimated GEV parameters that are estimated by fitting the collected extreme measurements when the bandwidth is high (B=H) for all servers. The SPT is associated with an individual monitoring session and its data can be updated periodically based on the collected extreme measurements. Table (4.1) shows a snapshot of the SPT.

**Table 4.1: Server Performance Table (SPT)**

<table>
<thead>
<tr>
<th>Server ID</th>
<th>B = H</th>
<th>GEV parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( W_i )</td>
<td>High Delay Extremes(s)</td>
</tr>
<tr>
<td>1</td>
<td>0.730</td>
<td>0.1029, … ,0.1037</td>
</tr>
<tr>
<td>2</td>
<td>0.476</td>
<td>0.1028, … ,0.1036</td>
</tr>
<tr>
<td>3</td>
<td>0.450</td>
<td>0.1036, … ,0.1042</td>
</tr>
<tr>
<td>4</td>
<td>0.554</td>
<td>0.1013, … ,0.1031</td>
</tr>
</tbody>
</table>
STEP 2:

Once the weights are determined, we utilize a Behavioral Assessment LOP Algorithm (BALOPA) to detect the MCs' behavior. First, the algorithm computes the GEV probabilities of a set of selected extreme measurements for every MC associated with the certain server. This process requires multiple SPTs, where each of them represents a snapshot of evaluation given monitoring session. Second, for each MC, all computed GEV probabilities are combined by using LOP. The constructed LOP probabilities reflect the best estimation of the MCs' behavior through the entire service evaluation process. The following pseudo-code describes the details of our BALOPA algorithm:

4.3.1. Behavioral Assessment LOP Algorithm (BALOPA)

**Input:** \( T_{SPT} = \) Set of SPT tables for multiple monitoring sessions

\( S_{ID} = \) Selected server ID

\( C_{MCs} = \) All MCs that are associated with the server \( S_{ID} \)

\( M_\theta = \) Array of extreme delay measurements for all MCs of a given server \( S_{ID} \)

**Output:** \( LOP_\theta = \) Array of calculated probabilities of extreme delay measurements using LOP for all MCs that are associated with server \( S_{ID} \)

---

**Step 1:** Let \( L_{SPT} = |T_{SPT}|: number of monitoring sessions of server \( S_{ID} \)

Let \( C_N = |C_{MCs}|: number of MCs that are associated with server \( S_{ID} \)

Let \( LOP_\theta = 0 \)

**Step 2:** For \( c = 1 \) To \( C_N \)
For $i = 1$ To $L_{SPT}$

$<\text{params}> = T_{SPT}[i][S_{ID}, \text{params}]$

// retrieve the GEV parameters and status weight of $S_{ID}$

$W_i = T_{SPT}[i][S_{ID}, W]$

$\text{LOP}_\theta[c] = \text{LOP}_\theta[c] + (W_i \times \text{GEV}<\text{params}, M_\theta[C_{MCs}[c]] > )$

// compute $\text{LOP}$ of $\theta$ using GEV

End for

End for

**Step 3**: End of Algorithm

**STEP 3**: In this step, a MCs discrimination process is applied to isolate well-behaving MCs (i.e., trusted clients) from misbehaving MCs (i.e., un-trusted clients). Towards this end, we introduced the Mobile Clients Fidelity Algorithm (MCFA) which utilizes the BALOPA algorithm to enable MCs behavioral recognition. The MCFA uses the calculated MCs' LOP probabilities to create initial clusters by using the K-means algorithm. The Sum of Squared Errors (SSE) [63] is then used to measure the distances between the centroids of the initial clusters. Two clusters with the least distance between their centroids are merged into a single cluster. This merging process is then repeated until it results in two clusters that represent the set of trusted and un-trusted MCs. SSE is defined as follows:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} D(c_i - x)^2$$

(4.3)
Where $D$ is the Euclidian distance function between centroid $c_i$ and a point $x$ in cluster $C_i$, and $K$ is the number of clusters. The final SSE value represents the split point between the two final clusters that represent the trusted and un-trusted MCs. In this work, we conducted prior training process to come up with the SSE value that signifies the best distinction point between trusted and un-trusted MCs clusters. The following pseudo-code illustrates the MCFA algorithm:

### 4.3.2. Mobile Clients Fidelity Algorithm (MCFA)

**Input:** $S_{ID}$ = Selected server ID

- $LOP_\theta$ = Array of calculated probabilities of extreme delay measurements of server $S_{ID}$

- $SSE_{split}$ = Sum of Squared Errors

- $T_{spt}$ = SPT of server $S_{ID}$

**Output:**

- $TMCs$ = Trusted MCs data

- $uTMCs$ = Un-trusted MCs data

- $T_{spt}$ = Updated version of $T_{spt}$

---

**Step 1:** Initialize $K$ clusters based on the computed $LOP_\theta$ probabilities using K-means, where

$$K \geq \frac{|LOP_\theta|}{ClusterDensity}$$

$$C = \text{Kmeans}( LOP_\theta, K)$$

Sort the clusters' centroids $C$

$$C_{\text{sorted}} = \text{Sort}(C, \text{“descending order”})$$
Step 2: Reassign and merge the constructed clusters by calculating SSE among clusters' centroids.

\[ C_k = |C_{sorted}| \]

\[ C_{pvt} = C_{sorted}[1] \]

cnt = 1

\[ C_{cnt} = C_k \]

While \((C_k > 2 \&\& \text{cnt} < C_{cnt})\) Do

\[ C_{error} = \sqrt{(C_{pvt} - C_{sorted}[\text{cnt}])^2} \]

If \((C_{error} < \text{SSE}_{split})\)

Merge \(C_{pvt}\) data with \(C_{sorted} [\text{cnt}]\) data and calculate new centroid,

\[ c_{data}^{\text{merge}} = \text{merge}(c_{pvt}^{\text{data}}, c_{sorted}^{\text{data}}[\text{cnt}]) \]

\[ C_{pvt} = \frac{\sum c_{mrg}^{\text{data}}}{|c_{mrg}^{\text{data}}|} \]

Decrement \(C_k\) by 1

End if

Increment \(\text{cnt}\) by 1

End while

\[ \text{TMCs} = c_{data}^{\text{mrg}} \]

\[ \text{uTMCs} = c_{sorted}^{\text{data}} \cap \text{TMCs} \]

Step 3: Compute new GEV for TMCs and update \(T_{spt}\) GEV parameters

\[ T_{spt}[\text{params}] = \text{GEV(TMCs)} \]

Step 4: End of Algorithm
4.4. Experimental Results

We employ the MATLAB-based simulator that has been developed in the first stage to verify the efficacy and robustness of our proposed approach. The simulator has been adjusted and modified to fit our proposed BALOPA and MCFA algorithms. The simulator runs in two phases, namely: the knowledge building phase and the on-line MC classification phase. In the first phase, the simulator calculates the best SSE that represents the discrimination point between trusted and un-trusted MCs. Through this process, the simulator investigates the behavior of the MCs that fluctuate between the trusted and un-trusted clusters, and measures the SSE between the centroids of the clusters over runtime. The second phase applies our proposed algorithms to the output of the first phase to distinguish between trusted and un-trusted MCs.

Fig. 4.1 shows that 14 MCs requested services from server number 1. The BALOPA and the MCFA algorithms were applied to compute probabilities of certain extreme delay measurements. The algorithms successfully classified the MCs into two distinct clusters based on their behavior. Based on the clustering results, the majority of MCs (trusted) showed a consistent behavior through the monitoring sessions and converged into one cluster, while the other MCs (un-trusted) did not report consistent measurements and were grouped into another cluster. Thus, the output of the clustering process identifies the extreme measurements reported by the trusted clients. Only this trusted data is then used by the service evaluation process to build up more accurate performance model of the monitored service. In the MCFA algorithm, we used SSE = 0.005 as the cluster isolation factor. Based on this factor, 11 of the MCs (trusted) that
reported extreme delay measurements when the bandwidth is high (B=H) have predicted degradation in server’s performance, while the other MCs (un-trusted) did not.

We applied the True Positive Ratio (TPR) [61] as a prediction sensitivity measurement to evaluate the performance of the MCFA output. The TPR equation uses True Positive (TP) and False Negative (FN) factors. The TP refers to the MCs LOP probability that has been predicated by the QoS model, and that matches the historical performance data. Whereas, the FN refers to the MCs LOP probability that has not been predicted, and that did not match the historical performance data. Fig. 4.2 shows the prediction ratio of the trusted MCs over all MCs. The output depicted in Fig. 4.2 demonstrates that the trusted MCs have captured the service degradation on servers 1, 4, 6, and 8, and the predication ratio was higher than if all MCs were used in the evaluation process. Therefore, adopting part of the data reported by the trusted MCs only enhances the accuracy of the service evaluation process.

![Figure 4.1: LOP Probabilities for 14 MCs Using Server 1](image)

55
Fig. 4.3 shows the number of the trusted MCs versus all the MCs involved in the service evaluation process of servers 1, 4, 6, and 8. Our finding demonstrates clearly that we can reduce the number of MCs involved in the service evaluation process while maintaining the same or even improving the accuracy level. Furthermore, reducing the number of MCs can decrease the evaluation process overhead.

Figure 4.2: TPR of Trusted MCs vs. All MCs for Servers 1, 4, 6, 8
In this stage of our research, we presented a new approach for improving the performance of client-based QoS service modeling by excluding MCs’ out-of-profile data from the service evaluation process. The proposed technique is based on two algorithms, namely: BALOPA and MCFA. These algorithms utilize LOP and GEV to sift the service performance information that can mislead the service assessment process and increase the service evaluation time. Moreover, our algorithms are capable of distinguishing and excluding the misbehaving MCs while maintaining or improving the service performance prediction. Our results demonstrate the efficiency of the proposed technique and algorithms.

**Figure 4.3: Number of Trusted MCs and All MCs Participating in Service Assessment Process for Servers 1, 4, 6, and 8**
CHAPTER 5
A CLIENT-BASED QOS APPROACH USING GENERALIZED EXTREME VALUE THEOREM IN MULTI-HOP NETWORK ENVIRONMENTS

5.1. Introduction

In most service assessment scenarios, service providers' efforts to provide reliable services can face a challenge in its continuous service evaluation process while managing Service Level Agreements (SLAs) for mobile clients. However, the challenge might fall into maintaining several levels of service assurance starting with utilizing a consistent network infrastructure and ending with hosting smart service techniques capable of operating in different network topologies.

In this chapter, we present a novel approach that evaluates service performance through collecting service measurements from mobile clients in a multi-hop network environment. The proposed approach utilizes a System of Linear Equations and Generalized Extreme Value Theorem techniques to predict network link performance degradation by estimating the delay extremes on each hop of a given network topology. Consequently, service performance can be evaluated and improved through a continuous assessment process of the network’s links behavior over time.

While our previous studies [28], [33] have focused on single-hop networks, the objective of this research stage is to take another leap towards studying client-side QoS assessment in a multi-hop network environment. In this research stage, we proposed an approach that employs a link-based GEV modeling to provide an accurate assessment of the network link delays. It assumes that the MCs report snapshot-based information that consists of delay extremes and hop-by-hop details of the end-to-end traffic load. That is, every snapshot provides all the delay extremes associated with a particular path for each
MC in the snapshot. On the long run, the captured snapshots are organized and solved as a system of linear equations to estimate link delays. Furthermore, the parameters of a set of GEV models are estimated to evaluate and predict link delay performance over time.

5.2. Problem Description and Modeling

Let’s assume that a network snapshot $S_{MCI} = (L_{MCi}, MCs)$ is a directed graph in which the link $l_{ui} \in L_{MCi}$ and $MC_i \in MCs$ is the $i^{th}$ mobile client which sends a request to a given dedicated service $SP$ in the given snapshot $S_{MCi}$. Figures 5.1a and 5.1b represent snapshots in single and multi-hop networks, respectively.

![Figure 5.1: (5.1.a) Single-Hop Network Topology Snapshot, (5.1.b) Multi-Hop Network Topology Snapshot](image-url)
The following provides definition of the terms used in our proposed approach:

- \( D_{MCS} \): The sequence of all delay extremes collected by the MCs in a given snapshot.

\[
D_{MCS} = [d_{MC_1}^k, d_{MC_2}^k, d_{MC_3}^k, \ldots, d_{MC_i}^k]
\]

- \( d_{MC_i}^k \): total delay of the path \( p_{lt} \)

\[
d_{MC_i}^k = \sum_{u=1}^{p_{lt}} l_u,
\]

Where, \( 1 \leq i \leq |MCs|, 1 \leq k \leq |P_{MC_i}|, |p_{lt}| = \text{number of links on path } p_{lt}

- \( P_{MC_i} \): The sequence of all paths used by MC\(i\) in a given snapshot, where \( k \) represents the path ID

\[
P_{MC_i} = [p_{l_1}, p_{l_2}, p_{l_3}, \ldots, p_{l_k}]
\]

- \( L_{MC_i} \): The sequence of all links used by \( MC_i \) in a given snapshot,

\[
L_{MC_i} = [l_1, l_2, l_3, \ldots, l_U], \text{Where } U \text{ represents the number of links in the snapshot and } 1 \leq U \leq |MCs \text{ paths links}|
\]

Arranging the information of the captured snapshots as a linear system of equations, \( A^{-1} \times X = B \), is straightforward since the links' paths and delay extremes can be organized in matrices as follows:

\[
\begin{bmatrix}
P_{1,U}^{MC_i} \\
P_{2,U}^{MC_i} \\
P_{3,U}^{MC_i} \\
\vdots \\
P_{W,U}^{MC_i}
\end{bmatrix}_{W \times U} \times 
\begin{bmatrix}
l_1 \\
l_2 \\
l_3 \\
\vdots \\
l_U
\end{bmatrix}_{U} = 
\begin{bmatrix}
d_{1,U}^{MC_i} \\
d_{2,U}^{MC_i} \\
d_{3,U}^{MC_i} \\
\vdots \\
d_{W,U}^{MC_i}
\end{bmatrix}_{W}
\]

Where \( W > U \)
5.3. The Proposed Algorithms

Overall, our proposed approach consists of two new algorithms, namely: the Link Delay-Driven Algorithm (LDDA) and the Link Delay Performance Assessment Algorithm (LDPAA).

5.3.1. Link Delay-Driven Algorithm (LDDA)

This Algorithm represents the first step of the service evaluation process. Since we assume that the evaluation process can be done based on time frames (snapshots), all retrieved information (delay extremes and paths’ information) has to be adequate to be represented and solved as a system of linear equations. To ensure that, the proposed algorithm runs through, collects snapshots iteratively and measures the Sum of Squared Error (SSE) among the calculated links' delay. The algorithm has to ensure that the SSE is minimized. The execution of the algorithm is terminated when successive calculations of the SSE metric do not result in a lower value. The SSE is given by [63]:

\[
SSE = \sum_{i=1}^{\vert R \vert} \sum_{\nu \in r^i_d} (l^\nu - (r^i_d))^2
\]  

(5.3)

\[ R = [r^1_d, r^2_d, ... r^H_d], \]
represents the sequence of unique paths for a given client MC_i, where \( 1 \leq H \leq |\text{Unique Paths}| \)

\[ r^i_d = \text{The delay extremes of link } L_{MC_i} \text{ that belong to the similar paths} \]

The following pseudo-code describes in details the LDDA algorithm:

**Link Delay-Driven Algorithm (LDDA)**

**Input:** \( S_{data} \) = Set of snapshots

\[ Trm_{THR} = \text{Algorithm Termination Threshold} \]
**Output:** $L_{Mci}^{value} = $ Estimated link delay extremes

---

**Step 1:** Retrieve all extremes of the first snapshot, such that $d_{Mci} = S_{data}[x]$ , where 

\[1 \leq x \leq |d_{Mci}|, x \text{ is snapshot size, and } 1 \leq i \leq |MCs|\]

Assign snapshot flag, $ValidSnapshot = true$

Assign Error control flag, $Prev_{error} = -1$

**Step 2:** Extract all possible links IDs from $d_{Mci}$

$L_{Mci} = MCsLinks(d_{Mci}) \cup L_{Mci}$ in form of $[l_1, l_2, \ldots, l_U]$

**Step 3:** Extract all possible paths from $d_{Mci}$

If ($ValidSnapshot == true$) then

\[P_{Mci} = MCsPaths(d_{Mci}) \cup P_{Mci} \text{ in form of } [P_{Mci}^1, P_{Mci}^2, \ldots, P_{Mci}^W]\]

Else

\[d_{Mci} = S_{data}[x] \cup S_d\]

\[P_{Mci} = MCsPaths(d_{Mci})\]

End if

**Step 4:** Calculate the snapshot links' delays

If ( $|P_{Mci}| > |L_{Mci}|$ )

\[L_{Mci}^{value} = (P_{Mci})^{-1} \times d_{Mci}\]

Else

$ValidSnapshot = false$

Goto Step 3

End if

**Step 5:** Calculate $SSE$ such that no similar successive $SSEs$ exceed $Trm_{THR}$
\[ Curr_{error} = SSE(L^{value}_{MC_l}, a_{MC_l}) \quad \text{// See Eq. (5.3)} \]

If \((Prev_{error} \neq Curr_{error})\) then

\[ t = 1 \]

Else

\[ t = t + 1 \]

End if

If \((t = Trm_{THR})\) Then Goto Step 6

Else

\[ Prev_{error} = Curr_{error} \]

ValidSnapshot = false

GotoStep 3

End if

**Step 6:** End of Algorithm

### 5.3.2. Link Delay Performance Assessment Algorithm (LDPAA)

In order to provide accurate service performance assessment, the LDPAA algorithm employs GEV models to estimate the network link delays. Every link has a group of estimated delays calculated from different snapshots over time by using the LDDA algorithm presented above. The BM method is then used to select extreme samples that represent the GEV distribution. The idea behind the Block Maxima technique is to divide the collected delay samples into \(m\) successive blocks of size \(n\) and electing the maximum value of each block. The elected maximum values represent the extremes of the observations. After obtaining the Block Maxima samples, the LDPAA algorithm chooses a suitable GEV model using distribution fitting and retrieve the parameters of the
selected GEV distribution. The resulting GEV models represent the links’ delay behavior throughout the service evaluation process. The following pseudo-code describes the details of our LDPAA algorithm:

**Link Delay Performance Assessment Algorithm (LDPAA)**

**Input**: \( FA \) = Number of simulation runs used to collect link delay extremes.

**Output**: \( P_{GEV} \) = GEV parameters for all network snapshot links

---

**Step 1**: \( L_{GEV} = [] \), Array that holds all link delay extremes for different snapshots

\( P_{GEV} = [] \), Array that holds GEV parameters of each link

**Step 2**: Collect different estimated link delay extremes for different runs by calling the LDDA

For \( row = 1 \) to \( FA \)

\[ L_{GEV}(row, :) = \text{delay extremes obtained from the LDDA} \]

End for

**Step 3**: Extract GEV parameters for each link

For \( col = 1 \) to \( \text{SizeOf}(L_{GEV}) \)

\[ [\kappa, \sigma, \mu] = \text{GEV}(L_{GEV}(:, col)) \]

\[ P_{GEV}(col, :) = [\kappa, \sigma, \mu] \]

End for

**Step 4**: End of Algorithm

5.4. Experimental Results

We collected QoS measurements in terms of delay extremes and hop-by-hop route information generated from 25 mobile clients requesting service from 8 different service
providers. We developed a MATLAB-based simulator to execute our algorithms in a mobile environment using the Random Waypoint Mobility model to simulate 25 MCs moving within an area of $(3000 \, m \times 3000 \, m)$ through several snapshots (i.e., monitoring sessions) [57]. The simulator is based on the EVIM software package [EVIM] to estimate the best GEV model that fits the collected measurements. The evaluation process is based on measurements collected from mobile clients $MC_2$ through $MC_5$ as depicted in Fig. 5.1.b.

The LDDA algorithm was applied to the captured snapshots in order to estimate the links’ delays. After 46 iterations, the algorithm converged to the estimated links’ delays. Fig. 5.2 shows the SSE values produced by the LDDA algorithm.

![Figure 5.2: SSE Values Produced by the LDDA Algorithm](image-url)
Table 5.1 shows the estimated links’ delays based on the topology depicted Fig. 5.1.b.

Table 5.2 shows the estimated links' delays for single and multi-hop paths for all requests that have been made by MC2 through MC5. It should be emphasized here that the total delay for multi-hop routes is calculated based on the delays of the underlying single-hop links that make up that route.

**Table 5.1: Estimated Links’ Delays of the Captured Snapshots**

<table>
<thead>
<tr>
<th>Link ID</th>
<th>Link Delay (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>l₁</td>
<td>0.045</td>
</tr>
<tr>
<td>l₂</td>
<td>0.157</td>
</tr>
<tr>
<td>l₃</td>
<td>0.112</td>
</tr>
<tr>
<td>l₄</td>
<td>0.189</td>
</tr>
<tr>
<td>l₅</td>
<td>0.077</td>
</tr>
<tr>
<td>l₆</td>
<td>0.391</td>
</tr>
<tr>
<td>l₇</td>
<td>0.157</td>
</tr>
<tr>
<td>l₈</td>
<td>0.202</td>
</tr>
</tbody>
</table>

**Table 5.2: Sample of Estimated Links’ Delays of Single and Multi-Hop Paths**

<table>
<thead>
<tr>
<th>Route ID</th>
<th>Multi-Hop Delay (s)</th>
<th>Single-Hop Delay (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(MC5 → SP) : l₇+l₁+l₃+l₅ ≈ 0.391</td>
<td>(MC5 → SP) = 0.391</td>
</tr>
<tr>
<td>2</td>
<td>(MC2 → SP) : l₁+l₃+l₅ ≈ 0.234</td>
<td>(MC2 → SP) = 0.234</td>
</tr>
<tr>
<td>3</td>
<td>(MC3 → SP) : l₃+l₅   ≈ 0.189</td>
<td>(MC3 → SP) = 0.189</td>
</tr>
<tr>
<td>4</td>
<td>(MC4 → SP) : l₅     ≈ 0.077</td>
<td>(MC4 → SP) = 0.077</td>
</tr>
</tbody>
</table>

To evaluate the links' delay performance, we applied the LDPAA algorithm on the collected measurements through all the monitoring sessions. Fig. 5.3 shows the extracted GEV probabilities for link (lᵢ) based on the data collected in different measurement snapshots. The figure demonstrates a significant improvement in the
extracted GEV probabilities as more measurements are collected. It should be emphasized that the measurements collected in snapshots 1 through \(i\) are augmented to the measurements collected in snapshot \(i+1\). Thus, snapshot \(i+1\) contains all the measurements collected in the prior snapshots.

**Figure 5.3: GEV Probabilities vs. Delay Extremes for Link \((l_1)\)**

In our second experiment, we focused on a scenario that calculates the delay from a mobile client (MC\(_5\) in this case) point of view. For each link, we measured the \(TPR\) (True Positive Ratio) based on the GEV model and the instantaneous link delay measurements. The \(TPR\) metric is calculated as follows:

\[
TPR = \frac{TP}{(TP + FN)} \tag{5.4}
\]
The TP (True Positive) refers to the number of correctly predicted link delays that appeared previously in the collected delay measurements. While FN (False Negative) refers to the number of mispredicted link delays that appeared previously in the collected delay measurements. The results demonstrate that the TPR of the individual network links can be improved as more measurements are collected. That is, providing more data in terms of delay extremes enhances the fitted GEV model, and that improves the prediction performance of link delay extremes.

Fig. 5.4 shows the prediction sensitivity (TPR) of all routes that used by MC₅. Moreover, Table 3 illustrates the details of all possible routes used by MC₅.

![Figure 5.4: True Positive Ratio (TPR) of Extreme Delays of All Routes Used by MC₅](image-url)
Moreover, we performed predication sensitivity of all routes (see Table 5.3) used by the mobile client MC₅. We applied the TPR metric for each route in the different snapshots. The results in Fig. 5.5 show the prediction performance of the delay extremes of all routes used by MC₅.

Table 5.3: All Possible Routes of MC₅

<table>
<thead>
<tr>
<th>Route ID</th>
<th>Route</th>
<th>Link ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MC₅ → MC₂ → MC₃ → MC₄ → SP</td>
<td>l₁₇+l₁₄+l₁₃+l₁₅</td>
</tr>
<tr>
<td>2</td>
<td>MC₅ → MC₃ → MC₄ → SP</td>
<td>l₁₆+l₁₄+l₁₃</td>
</tr>
<tr>
<td>3</td>
<td>MC₅ → MC₂ → MC₄ → SP</td>
<td>l₁₇+l₁₂+l₁₅</td>
</tr>
<tr>
<td>4</td>
<td>MC₅ → MC₂ → MC₃ → SP</td>
<td>l₁₇+l₁₁+l₁₄</td>
</tr>
<tr>
<td>5</td>
<td>MC₅ → MC₃ → SP</td>
<td>l₁₆+l₁₄</td>
</tr>
<tr>
<td>6</td>
<td>MC₅ → SP</td>
<td>l₁₈</td>
</tr>
</tbody>
</table>

Figure 5.5: Prediction Sensitivity of the Delay Extremes of all Routes that Originate from MC₅ (based on the topology described in Figure 5.1.b)
Fig. 5.5 illustrates the benefit of collecting more measurements before constructing the GEV model. This indicates that the prediction performance can be improved as more data is used to construct the GEV model. It should be noted here that our proposed LDDA terminates when the algorithm finds that collecting more data will not help to improve the quality of the resulting GEV model (i.e., SSE is minimized).

The outcomes of our current approach have shown that the prediction performance of link delays in multi-hop networks can be improved by adopting our proposed algorithms. The proposed approach utilizes a SLE and GEV techniques to predict network link performance degradation by estimating the delay extremes on each hop of a given network topology. The proposed algorithms, namely LDDA and LDPAA, are used to estimate the links’ delays and produce accurate GEV models that represent the collected measurements. The results demonstrate the efficiency of the proposed approach and algorithms in terms of prediction sensitivity (true positive ratio).
CHAPTER 6

TOWARDS A CLIENT-SIDE QOS MONITORING AND ASSESSMENT USING GENERALIZED PARETO DISTRIBUTION IN A CLOUD-BASED ENVIRONMENT

6.1. Introduction

Recently, cloud computing and cloud-based services have been gaining a lot of momentum due to their cost effectiveness. Cloud computing can be defined in many different ways, but the most realistic cloud computing definition that is related to our study is a vast resource pool that can be used to provide resources in a timely and reliable manner as demanded by the customers. Several researchers have started to study network issues that are stemming from the interaction between cloud clients and cloud service providers [64][65][66].

Assuring QoS guarantees to the cloud’s mobile clients is an important and unavoidable problem especially for delay sensitive application like VoIP. QoS is one of the challenging issues that can be negatively impacted by the lack of intelligent management and provisioning of the cloud’s resources. Currently, most QoS monitoring techniques have been based on service measurements collected by network elements (i.e., network-side monitoring) to evaluate the cloud’s performance.

Cloud Service Providers (CSPs) adopt different strategies to monitor and manage their cloud infrastructure and services. Even though they utilize the necessary tools to monitor and assess the performance of the provided services, these tools do not enable the clients to participate in the service monitoring and evaluation process. We believe that the clients should be involved in the performance monitoring and evaluation process and should assess whether the cloud service provider is abiding by the underlying SLA
In this chapter, we propose a novel client-based QoS approach for the early detection of cloud service degradations. The proposed algorithms employ the GPD approach to converge to a reliable QoS model based on collected delay measurements from the cloud’s mobile clients. Furthermore, by applying a data aggregation process, the proposed approach is capable of providing multi-level service performance assessment through analyzing the collected extreme measurements from VMs, zones and datacenters.

6.2. The Proposed Approach

In this approach, we propose a cloud-based architecture that delegates the monitoring role to the Mobile Clients (MCs). The mobile clients will be fully supervised by the Cloud Broker Manager (CBM). The Cloud Broker Manager advises the mobile clients to gather and forward network measurements (delay, bandwidth, etc.)—which we call extremes—when these measurements exceed a pre-defined threshold. In this work, we assume that the MCs and the cloud system (data centers, clusters and virtual machines) communicate though end-to-end transport layer connections (e.g., TCP and UDP). Also, the CBM can be implemented on different distributed environments such as cloud environments, GENI platform, and Hadoop/MapReduce framework. The primary role of the CBM is to utilize GPD to build a probability distribution based on the collected extremes. Furthermore, the CBM reports summaries of the performance evaluation process to the CSPs. The following provides details of the different modules of the proposed architecture:

- **GPD Model Builder**: fits the GPD model to the new collected extremes, and tunes up the readily constructed distributions to fit to the upcoming extremes;
• **Extremes Aggregate Manager**: Aggregates the collected VMs’ extremes through level-based aggregation procedure (i.e., VM-level, cluster-level and datacenter-level) as requested by the CBM;

• **Extremes DB**: Hosts the collected VMs’ extremes for short and long-term evaluation;

• **Monitor Manager**: Coordinates with the CBM to receive the updated monitoring criteria (thresholds) and extremes and forwards the collected extremes to the Extremes DB;

• **Shadow DB**: Saves the POT threshold temporarily on the client-side to minimize the need to communicate measurements that do not exceed the given POT;

• **Cloud Manager**: Controls and manages the Cloud resources, and interacts with the MCs and the CBM regarding the services’ performance;

• **Provision Pool**: Contains resources as well as VMs that are available to support the MCs requests;

• **Active Pool**: Holds all VMs that are running to serve the MCs requests;

• **Migration Pool**: hosts all VMs that have been suspended due to cost considerations or due to unexpected malfunctions;

In our proposed client-side monitoring system, the MCs interact with the CSP to provision VMs capable of providing the requested services. Hence, the CSP initiates the VMs by allocating the necessary resources demanded by MCs, which can then be made available in the *provision pool*. The VMs that are initiated successfully and made ready to migrate to the *active pool*, and can be accessed by the MCs. Accordingly, the CSP notifies the MCs to direct their requests to the newly activated VMs. Also, the MCs are
notified to start the monitoring process and collect the network measurements. The monitoring process begins when the MCs receive threshold values forwarded by the CBM to measure the service performance of the active VMs. In the process, the MCs forward all extremes to the CBM when they exceed these thresholds and do not match the extremes that are stored in the shadow DB. It should be emphasized here that the major role of the Shadow DB is to keep the fitted GPD models up-to-date without overwhelming the CBM to recalculate newer GPDs when new extremes are received. Thus, MCs forward only those extremes that might change the fitted GPD models. Furthermore, the Shadow DB keeps the constructed GPD models tuned up by updating the GPD’s POT values. The Shadow DB synchronizes with the associated GPD’s POT values that are stored in the CBM’s extremes DB whenever newer extremes are received from the MCs. The Shadow DB can be initiated and operated temporarily while the MCs are active and that does not prevent the proposed architecture to scalable.

Upon receiving a request from a given CSP to conduct a certain level of service evaluation, the CBM retrieves the related extremes from the Extremes DB and sends them to the GPD model builder. An appropriate GPD model is constructed and then applied to the received service assessment request. The probability with maximum value represents the best prediction of service performance for a given request. In order to provide a multi-level service performance analysis, an aggregation process can be applied to the collected VMs extremes. Through this process, every extreme that belongs to the same zone or to the same data center can be aggregated into one data set, and then a suitable GPD model can be constructed accordingly.
The service evaluation process conducted by the CBM provides feedback to the cloud service provider that shows the current and predicted performance of the service provided by the active VMs. However, any VM that suffers from performance issues will be allocated to the Migration Pool. The CSP can then react to the Broker evolution by performing an extensive performance evaluation and providing necessary tuning to the offered service. This tuning can be achieved in the Migration Pool by changing the number of resources that are dedicated to the service, changing the physical location of the service, or by applying a proactive technique (e.g., load balancing).

Fig. 6.1 depicts the main components that comprise our proposed client-side performance evaluation model.
6.2.1. On-Time Cloud Service Assessment Algorithm (OTCSA)

In order to capture the immediate VMs’ performance, the proposed client-side performance evaluation system employs the On-Time Cloud Service Assessment Algorithm (OTCSA) as a core part of its underlying architecture. The OTCSA detects the
VMs’ performance degradations and fluctuations by applying a process that combines the process of extracting a suitable GPD model based on the collected VMs’ extreme measurements with a pattern matching technique. First, after applying the MRL and PET techniques to the collected VMs’ extremes, an adequate lower threshold is calculated, and a GPD model is fitted to the extremes over the calculated threshold. Second, the OTCSA uses the extracted GPD model to calculate the quantile estimate probability and confidence interval for high extremes above the threshold. Furthermore, the proposed algorithm constructs lower and estimated bound values based on a given confidence level. These bounds are used in the pattern matching process to identify the VMs that generate extreme measurements within $M_{err}$ of the estimated bound value. Thus, the pattern matching process measures the deviation of the VMs’ extreme measurements from the overall extreme measurements collected within the monitoring session by calculating the VM Scoring Ratio (VSR). The VSR can be defined as follows:

$$\text{VSR} = \frac{\sum(|E - P_E| \leq M_{err})}{\sum(|E - P_L| \leq M_{err}) + \sum(|E - P_E| \leq M_{err})}$$  \hspace{1cm} (6.1)

Where, $E$ represents VM extremes, $M_{err}$ represents a predefined matching error value, and $P_L$ and $P_E$ represent the Lower and Estimated bound values, respectively.

In the final stage of the pattern matching process, the proposed algorithm uses the calculated VSR to categorize the VMs into two groups, namely: Critical Region (CR) and Safe Region (SR) groups. The CR group includes the VMs that show degradation in their service and need instant care from the CSP. On the other hand, all VMs that belong to the
SR show an acceptable service status. The CBM frequently reports feedbacks to the CSP regarding the CR’s VMs.

The following pseudocode describes the details of the proposed OTSCA algorithm.

**On-Time Cloud Service Assessment Algorithm (OTCSA)**

**Input:**

\[ VM_i \]: Virtual Machine ID

\[ E \]: VM_i’s Extremes

\[ tp \]: Tail probability

\[ ci \]: Confidence Interval

\[ Err \]: Predefined value that measures the difference between the extremes and the matching patterns.

\[ R \]: isolation factor that is used to categorize VMs

**Output:**

\[ CR, SR \]: Critical and Safe Region Sets

---

**Step 1:** Calculate a threshold \( u \) using MRL and PET such that it satisfies the best GPD approximation.

\[ u = \min \{ \text{MRL}(E), \text{PET}(E) \} \]

**Step 2:** Retrieve all extremes that are above \( u \), and extract a suitable GPD model by estimating the shape and scale parameters.

\[ E_{POT} = \{ \forall E(i) \geq u \}, 1 \leq i \leq |E| \]

\[ <\xi, \beta> = \text{GPD}(E_{POT}) \]
Step 3: Calculate quantile estimate and confidence interval for the $E_{POT}$ and extract lower and estimated bound pattern values $[P_L, P_E]$, respectively.

$$[P_L, P_E] = \text{GPD-Quantile} (\xi, \beta, \ u, \ E_{POT}, \ \text{tail\_probability}, \ \text{"Approx = likelihood"}, \ \text{Confidence\_Interval})$$

Step 4: Calculate VMs Scoring Ratio (VSR) by applying a pattern matching process.

Initialize a counter variable, $n = 1$.

For $\forall LEU \in [P_L, P_E]$ Do

$$W_{VM}(n) = \sum_{e=1}^{[E]} |E(e) - LEU| \geq Err$$

$n = n + 1$

End For

$$\text{VSR} = \frac{\sum_{i=1}^{W_{VM}(2)} W_{VM}(i)}{\sum_{i=1}^{W_{VM}(1)} W_{VM}(i)}$$

Step 5: Update the Critical Region (CR) and the Safe Region (SR) sets.

If $\text{VSR} > R$ Then

$$CR = CR \cup VM_i$$

Else

$$SR = SR \cup VM_i$$

End If

Step 6: End of Algorithm

6.3. Experimental Results

Our experiments were conducted on a cloud-based system that contains two simulated datacenters located in the USA and Europe, respectively. Each simulated datacenter contained one zone of six hosts (see Fig 6.2). Our experiments utilized CloudStack as the cloud manager and the XenServer Hypervisor [67] on each host.
Furthermore, six VMs have been deployed on dedicated hosts in each of the simulated datacenters. A MATLAB-based software has been developed and used in conjunction with the Pingdom web service [59] to collect and aggregate network measurements (delay and bandwidth extremes) from active VMs. We implemented the OTCSA algorithm in our software to build GPD models and evaluate the service performance of the active VMs.

The first experiment was conducted by applying the OTCSA to the extreme measurements that were collected from VMs. The proposed algorithm in its first stage extracted a threshold $u = 0.58$ sec after applying the PET and MRL techniques. Figures 6.3.a. and 6.3.b. show the MRL and PET of VMs, respectively.
Figures 6.3.a and 6.3.b show that both MRL and PET have approximated $u$ to be the same value. It should be emphasized here that the shape and scale parameters have to satisfy the stability state, in order to consider $u$ as a good threshold selection (see Fig. 6.3.b).

Figure 6.3: (6.3.a) MRL Plot of VM$_5$, (6.3.b) PET Plot of VM$_5$
In the OTCSA second step, a suitable GPD model was fitted to the extremes over the threshold $u$, and shape and scale parameters were estimated. Furthermore, the function $GPD$-$Quantile$ calculated $P_L = 0.66044$ sec and $P_E = 0.78153$ sec using $tail$-$probability = 0.98$ and confidence level = 0.98. Fig. 6.4 illustrates the lower and estimated bound values ($P_L$ and $P_E$) based on the extreme measurements collected from VM$_5$.

![Figure 6.4: Lower and Estimated Bound Values ($P_L$ and $P_E$) for VM$_5$](image)

The proposed algorithm performed the pattern matching process on the extreme measurements collected from VM$_5$ and calculated VSR = 0.5 and 0.69 for monitoring sessions 1 and 2, respectively. During monitoring session 2, we intentionally degraded the performance of VM$_5$ by significantly increasing the offered load to its hosted services. The extracted VSR values indicate that the VM$_5$ suffers from performance
degradation and the OTCSA grouped it within the CR set. On the other hand, in monitoring session 3, our algorithm calculated VSR = 0.09 and classified VM₅ to have a typical service performance; thus, excluding it from the CR set (joining the SR set). During monitoring session 3, we did not degrade the performance on VM₅ and that is reflected in the VSR value.

Overall, we conducted three monitoring sessions to evaluate the performance of the active VMs. In all runs of the OTCSA algorithm, we used $R = 0.5$ as the group isolation factor. Based on this factor, ten of the active VMs were analyzed. In the second monitoring session, our results illustrated that VM₈ and VM₉ were classified to have degraded performance, whereas, in the third session, both VMs recovered in terms of their performance. Fig. 6.5 depicts the CR and SR sets that resulted from the three monitoring sessions.

Another experiment involved the aggregation of the active VMs’ extreme measurements that belonged to the first simulated datacenter. We fitted two GPD models: the first model considered both CR and SR VMs’ extreme measurements while the second model only considered CR VM’s extreme measurements. Our findings demonstrate that excluding the SR VMs’ extreme measurements from the service assessment process allows us to construct more accurate GPD model with higher probabilities used to predict degraded performance. Fig. 6.6 illustrates the GPD model fitting outcomes.
Through this chapter, we presented a novel client-based service monitoring and evaluation approach for cloud-based services and infrastructure.

Our proposed approach relies on collecting and aggregating extreme measurements (delay and bandwidth extremes) from mobile clients that request services...
from a cloud platform. The primary objective of the proposed approach is to proactively identify cloud services and infrastructure with degraded performance by utilizing the GPD model. Our experimental results demonstrate that utilizing a GPD model enables us to detect degraded performance timely. Our experiments illustrate promising outcomes and demonstrate the efficacy of the proposed approach.
CHAPTER 7

AN ADAPTIVE CLIENT-SIDE QOS SERVICE ASSESSMENT APPROACH USING SOCIAL NETWORK ANALYSIS AND EXTREME VALUE THEOREM

7.1. Introduction

Delivering assured performance to the mobile clients of cloud-hosted services is a challenging problem. Nowadays researchers and practitioners are paying much attention to deliver performance optimized cloud services to mobile users. Towards this end, service analytics provides the insight that enables service providers to assess the performance of their offerings and take actions to increase customer satisfaction. However, and as we stated before, a broad number of research studies have undertaken the service evaluation process from one side; that is, the service-side perspective [68]. Conversely, clients’ assessment to the service has been mostly neglected.

Through this chapter, we propose a client-based QoS as a new and alternative approach to assess the performance of networked and cloud-based services. We exploit the SNA techniques in conjunction with GPD to evaluate the service performance from clients’ point of view. Thus, we developed a novel SNA-based clustering algorithm that analyzes the strength of the interconnection links between MCs and cluster related MCs in communities of similar behavior. The proposed algorithm proactively identifies clients that exhibit similar behavior through the Kendall-Tau statistic. The presented approach is effective in providing service providers with a better assessment tool to evaluate and improve their service offerings.

The following sections describe in detail the SNA principle and the advantages that empower the proposed approach. The client-based QoS Monitoring Architecture and Integer Linear Programming (ILP) formulation of the problem are also presented.
7.2. Social Network Analysis (SNA)

SNA can be defined as an approach that discovers and extracts meaningful information concluded from the users’ relationships embedded inside the social network. This information can be detected by measuring the strength of ties (connections) among users [44] [69] [70].

From the computer network point of view, the SNA has great potential to enhance the performance of the network infrastructure through analyzing the network measurements that collected from the monitoring process. It can be applied to improve the network reliably, security as well as marketing [71].

The simplest SNA model consists of nodes (actors) and links (relationships) that represent the flow among nodes[72]. Furthermore, the SNA has devoted techniques, such as centrality (degree, betweenness, closeness, and eigenvector centrality) to divide network nodes into subgroups such that each subgroup or cluster reflects specific relation pattern among certain nodes. Indeed, finding such patterns can provide valuable information that leads to a better understanding of the nodes’ relationships and the network’s structure [73].

From SNA’s prospective, There are two types of network clustering techniques; namely, community structure and social positions based clustering [44]. The community structure clustering can be determined based on a graph topology. That is, “clustered nodes are those tensely intra-connected in the graph structure while some loosely inter-connected nodes locate between clusters” [74]. On the other hand, the social-positions clustering approach classifies the nodes based on the similarity of their connections as well as specific common patterns such as similar neighbors [44].
In our research, we adopt the social-positions clustering to group MCs, the following sections present our SNA-based algorithms and techniques.

7.3. The Proposed Approach

Our proposed approach exploits the attributes of SNA and presents an efficient strategy that can be employed to improve the service monitoring process and lower its overhead. It is a step towards building a service performance predication model that combine SNA and GPD. More specifically, our approach is based on collecting MCs measurements while they are interacting with the offered services. The collected measurements can then be used as an effective indicator of MCs’ behavior if we consider certain conditions such as, MCs’ locations and MCs service type.

The first stage of the proposed approach explores the correlation among active nodes (MCs). This process enables us to identify and differentiate the MCs based on their behaviors. More specifically, we attempt to measure the correlation between each pair of MCs by using the Kendall Tau statistic based on the collected nodes measurements. Each link that connects the randomly elected pair is ranked by a weight based on the Kendall Tau computation. Thus, the rank value reflects the strength of the ties between the elected pair. Since the Kendall Tau correlation coefficient falls between [-1, 0, 1], we only consider the values that are greater than or equal to zero. That is, 1 represents a strong tie, whereas the absence of a tie is 0, and -1 represents a negative tie. However, we neglect the negative correlations between nodes since we assume that there are no negative behaviors amongst the nodes. The more positive value we can acquire from Kendall Tau computation the more behavior closeness we can determine of the elected pair. The Kendall Tau statistic is defined as follows [75] [76].
\[ \tau_B = \frac{\sum_{i<j}[S[MC_k(i) - MC_k(j)] \times S[MC_L(i) - MC_L(j)]]}{\sqrt{(n_0 - n_1)(n_0 - n_2)}} \]  

(7.1)

\[ n_0 = \frac{n(n - 1)}{2} \]  

(7.2)

\[ n_1 = \sum_i t_i(t_i - 1) \]  

(7.3)

\[ n_2 = \sum_j u_j(u_j - 1) \]  

(7.4)

\[ S(\nu) = \begin{cases} 
1 & \text{if } \nu > 0 \\
0 & \text{if } \nu = 0 \\
-1 & \text{if } \nu < 0 
\end{cases} \]

\[ n_0 = \text{Total number of possible pairs} \]

\[ t_i = \text{Number of tied } MC_k \text{ values in the } i^{th} \text{ group of tied } MC_k \text{ values} \]

\[ u_j = \text{Number of tied } MC_L \text{ values in the } j^{th} \text{ group of tied } MC_L \text{ values} \]

\( MC_k, MC_L = \text{any two mobile clients} \)

### 7.3.1. Problem Description and Formulation

Through this section, we focus our attention on formulating the proposed nodes’ behavior clustering problem using integer linear programming (ILP) [77][78]. The proposed formulation can be applied to small-scale networks to construct optimal clusters that contain nodes with similar behavior based on the Kendall Tau statistic. Our objective is to maximize the cluster size such that the nodes with strong connections can be
grouped and isolated from those with weak connections. The proposed ILP model has constants, constraints, and an objective function that can be described as follows:

- **Constants**:
  
  \( N \): Number of nodes
  
  \( \tau : N \times N \) input matrix where element \( \tau_{ij} \) represents the Kendall Tau statistic calculated based on the extreme delay measurements collected between node \( i \) and node \( j \)
  
  \[ 0 \leq |\tau_{ij}| \leq 1 \]

- **Variables**:

  \( X : N \times N \) output matrix, such that

  \[ x_{ij} = \begin{cases} 1, & \text{if node } i \text{ is clustered in cluster } j \\ 0, & \text{Otherwise} \end{cases} \]

- **Constraints**:

  \[
  \sum_{j=1}^{N} x_{ij} = 1, \quad \forall \ 1 \leq i \leq N
  \]

  \[
  0 \leq x_{ij} \leq 1, \quad \forall \ 1 \leq i, j \leq N
  \]

  And \( x_{ij} \) is binary

- **Objective Function**:

  \[
  \text{Max} \left[ 3 \times \sum_{k=1}^{N} \sum_{j=1}^{N} \sum_{i=1}^{N} \tau_{ij} \times y^k_{ij} - \sum_{k=1}^{N} \sum_{j=1}^{N} \sum_{i=1}^{N} \tau_{ij} \times (-x_{ik} - x_{jk}) \right]
  \]

  Where, \( y^k_{ij} \) is a binary ancillary variable, such that
\[
\begin{align*}
    y_{ij}^k & \leq x_{ik} \\
    y_{ij}^k & \leq x_{jk} \\
    y_{ij}^k & \geq x_{ik} + x_{jk} - 1 \\
    0 & \leq y_{ij}^k \leq 1
\end{align*}
\]

7.3.2. Extreme Social Bond Clustering Heuristic (ESBCH)

Once the weights are determined through calculating the Kendall Tau coefficients for the networks’ nodes, our proposed ESBCH can be applied.

The Node Bond Factor (NBF) metric can be formalized as follows:

\[
NBF = \frac{\sum_{l_{n_i} \in \mathcal{E}_n} l_{n_i}}{D_{n_i}}
\]

Where, \( n_i \) is any selected node in the network, \( D_{n_i} \) is the degree centrality of node \( n_i \), \( l_{n_i} \) is a link from \( n_i \) to any other node in the network. Figure 7.1 provides the detailed pseudo-code of our proposed ESBCH approach.

**Extreme Social Bond Clustering Heuristic (ESBCH)**

**Input**: \( G = \) weighted network

**Output**: Network with clustered nodes

**Step 1**: Computes Node Bond Factor (NBF) of \( G \)

\[ NR = NBF (G); \]

**Step 2**: Choose two nodes that have highest \( NBF \) scores

\[ NMax1 = Max(NR); \]

\[ NMax2 = Max(NR - NMax1); \]

**Step 3**: Find the shortest path between \( NMax1 \) and \( NMax2 \), and remove the link that has the least weight
\[ P_{short} = \text{ShortPath} \left( NMax1, NMax2 \right) \]

If \( P_{short} == 0 \) then goto Step 4

Else

Remove the weakest link of \( P_{short} \),

Goto Step 1.

**Step 4**: End of Algorithm

---

**Figure 7.1**: Illustration of the Clustering Process Used in the ESBCH
7.3.3. Immediate Service Performance Assessment Algorithm (iSPA)

The clusters’ extremes that resulted from ESBCH have to be evaluated in order to analyze and predict the service performance for potential degradations. We propose the Immediate Service Performance Algorithm to construct GPD models that fit the collected clusters’ extremes. Through this algorithm, we apply the MRL and PET to come up with the best threshold calculation that can be used to identify the most influential extremes. The GPD model then can be fitted to the extremes that fall over the calculated threshold. We need to mention here the constructed GPD models represent the core of service evaluation and prediction process.

The following provides the detailed pseudo-code of our proposed iSPA algorithm:

**Immediate Service Performance Assessment Algorithm (iSPA)**

**Input:** $E_{\text{extreme}} =$ Cluster’ Extremes

**Output:** $\xi, \beta =$ GPD model parameters

---

**Step 1:** Calculate threshold $u$ using MRL and PET such that it satisfies the best GPD approximation.

$$ u = \min \left[ \text{MRL} \left( E_{\text{extreme}} \right), \text{PET} \left( E_{\text{extreme}} \right) \right] $$

**Step 2:** Retrieve all extremes that are above $u$, and extract a suitable GPD model by estimating the shape and scale parameters.

$$ E_{\text{POT}} = \{ \forall E_{\text{extreme}}(i) \geq u \} , 1 \leq i \leq |E_{\text{extreme}}| $$

$$ < \xi, \beta > = \text{GPD} \left( E_{\text{POT}} \right) $$

**Step 3:** End of Algorithm
7.4. Experimental Results

The following experiments have been conducted using the Seattle Global Environment for Network Innovations (GENI) testbed [79]. GENI provides hardware and software infrastructure that enable the creation of at-scale network experiments with deeply programmable network elements. SeattleGENI is a community-driven large deployment network that utilizes resources denoted by users and service offering foundations such as computer network institutions, universities, and network service providers. It can operate as a part of user’s system with minimum consumption of user’s system resources and security since its programs only operate inside of a sandbox. Also, it compatible with various platforms and can be work on Windows, Linux, Mac OS, and portable devices platform. We allocated 105 and 150 vessels (MCs) from SeattleGENI testbeds connected through WAN resources distributed over USA, Europe, and Asia to collect measurements for analysis purposes. For the sake of performing reliable service assessment, we developed a MATLAB-based software that utilizes EVIM [55], Gephi [80], and UCINET [81] tools to analyze the collected extreme measurements.

7.4.1. Methodology

In order to perform the network measurement collection process, we developed python programs that monitor and manage the allocated vessels using UDP protocol through Single Virtual network Host (SVH) located in Kalamazoo-Michigan. The SVH runs a customized python program developed by SeattleGENI to operate as experiment manager and interact with the reserved vessels to collect network measurements. On the vessels side, we installed on each vessel our developed RestrictedPython code to react to the requests sent from SVH and from other reserved neighboring vessels in the network.
Additionally, the deployed code can be used for packet forwarding. We examined the performance of the allocated vessels by monitoring the behavior of the deployed python code through SVH web portal. The SVH web portal shows the collected network measurements (delays) from all allocated vessels over different run-time sessions. For the sake of extracting the delay extremes from the collected measurements, we applied MRL and PET techniques to compute the appropriate threshold \( u \) which then can be used in the fitting and modeling processes. From the extracted extremes, the strength of vessels’ relationships can be constructed by using the Kendall Tau statistic for each pair of nodes through the entire topology. A SNA-based heuristic (ESBCH) has been proposed to analyze and discover the relationships between the dedicated vessels. Since the relationships have been discovered, the proposed heuristic clusters the network nodes into distinct clusters.

7.4.2. Results

In our first experiment, the ESBCH has been applied on 105 vessels, and successfully split its nodes into two distinct clusters (31 nodes in cluster 1, and 44 nodes in cluster 2). The nodes that did not achieve strong bonds through the clustering process can be isolated and treated as independent nodes in the evaluation process. Figure 7.2 shows the ESBCH outcomes for the 105 vessels scenario.

For each created cluster, a GPD model is fitted based on the cluster’s extreme measurements. We have applied Goodness-of-fit to assure the accuracy of the fitted GPD for both the general GPD model, which is generated based on collected measurements from all the nodes, and the GPD models of cluster 1 and 2. Table 7.1 and Figure 7.3 show that the GPD models of cluster 1 and 2 were more accurate when compared with the
general GPD model. The accuracy can be distinguished from the variance of the calculated test values. Hence, if the computed test statistic is large, then the observed and expected values during the test are not close, and the model is poor fit for the data.

Table 7.1: Goodness of Fit (Chi-Squared) Test for 105 Mobile Clients

<table>
<thead>
<tr>
<th>GPD Model</th>
<th>Critical Value</th>
<th>α</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td></td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.01</td>
</tr>
<tr>
<td>Cluster 1</td>
<td></td>
<td>7.2893</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9.2364</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13.388</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.086</td>
</tr>
<tr>
<td>Cluster 2</td>
<td></td>
<td>5.9886</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.7794</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9.4877</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.668</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13.277</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.6416</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6.2514</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.8147</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9.8374</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.345</td>
</tr>
</tbody>
</table>

Figure 7.2: ESBCH Clustering Outcomes for 105 MCs
In order to verify the usefulness of adopting individual GPD models for the service evaluation process, we have examined the fitted GPD models on different scales of extremes for both the general and cluster 1 and 2 GPD models. Our finding demonstrated that clusters’ models have provided better fit to the data compared with that of the general model. That is, focusing on the measurements collected from MCs in a certain cluster make the fitting process more reliable, and significantly affects the results. Furthermore, this can be seen clearly from the calculated probabilities and relative frequencies of the general and cluster 1 and 2 models. Hence, the extracted empirical values (relative frequencies) were close to those of GPD models in both the general and cluster 1 and 2, and the matching error between any two values is small. Figures 7.4-7.5 and Tables 7.2-7.3 demonstrate the outcomes of GPD models and relative frequencies of the general and cluster 1 and 2 models.
Figure 7.4: The GPD and Calculated Relative Frequencies of the General and Cluster 1 Models

Figure 7.5: The GPD and Calculated Relative Frequencies of the General and Cluster 2 Models
Another Goodness of Fit test was applied to measure the accuracy of the constructed GPD models for each cluster that was generated from the ESBCH as well as the GPD of the general model. Figure 7.6 and Table 7.4 illustrate that clusters 1-3 GPD models have received lower test scores compared to the test score of the general GPD model. That is, the cluster models have achieved better fit to the collected measurements when compared with the general model. This provides us with a good insight about the efficiency of the proposed heuristic to classify the MCs into proper clusters.
and it successfully generated 3 distinct clusters. Figure 7.7 shows that the algorithm has

We have conducted a second experiment to cluster and evaluate 150 vessels. Again, our objective is to extract clusters with specific number of nodes such that each group of nodes has to demonstrate a similar behavior through their connections, and predict the service quality through employing GPD models of the clustered nodes on the service assessment process. Accordingly, the ESBCH was applied to cluster 150 MCs, and it successfully generated 3 distinct clusters. Figure 7.7 shows that the algorithm has
grouped the majority of the nodes into cluster 3. Furthermore, certain nodes have dominated the cluster since they achieved high degree-centrality, and that can be seen in clusters 2 and 3, respectively. The dominated nodes can provide exclusive information that might impact the service evaluation process.

Comparing the outcomes of the clusters 1-3 GPD models with the GPD general model in the second experiment, the constructed probabilities and frequencies were promising. Figures 7.8-7.10 and Tables 7.5-7.7 illustrate clearly the deviation of the estimated probabilities and frequencies of cluster 1 and the general GPD models. Due to the estimated POT threshold values of each GPD model, the calculated probabilities can be affected by the number of extremes that fall over the POT. Thus, increasing the estimated probabilities will lead to higher chances to predict the occurrence of upcoming

Figure 7.7: ESBCH Clustering Outcomes for 150 MCs
extremes from the fitted GPD models for each MC. Indeed, the accuracy of the fitted GPD model can impact the reliability of the service evaluation process.

Figure 7.8: The GPD and Calculated Relative Frequencies of the General and Cluster 1 Models

Table 7.5: Relative Frequencies and GPD Probabilities of the General and Cluster 1 Models

<table>
<thead>
<tr>
<th>Data Extremes</th>
<th>GPD of Cluster 1 Extremes</th>
<th>GPD of All Extremes</th>
<th>Cluster 1 Freq.</th>
<th>All Extremes Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>0.062418587</td>
<td>0.061726732</td>
<td>0.081</td>
<td>0.045</td>
</tr>
<tr>
<td>4.4</td>
<td>0.051318294</td>
<td>0.037054194</td>
<td>0.070</td>
<td>0.025</td>
</tr>
<tr>
<td>4.8</td>
<td>0.122973218</td>
<td>0.110455934</td>
<td>0.100</td>
<td>0.080</td>
</tr>
<tr>
<td>6</td>
<td>0.113510000</td>
<td>0.030455122</td>
<td>0.100</td>
<td>0.025</td>
</tr>
<tr>
<td>6.4</td>
<td>0.057773000</td>
<td>0.013165434</td>
<td>0.075</td>
<td>0.025</td>
</tr>
</tbody>
</table>
Figure 7.9: The GPD and Calculated Relative Frequencies of the General and Cluster 2 Models

Table 7.6: Relative Frequencies and GPD Probabilities of the General and Cluster 2 Models

<table>
<thead>
<tr>
<th>Data Extremes</th>
<th>GPD of Cluster 2 Extremes</th>
<th>GPD of All Extremes</th>
<th>Cluster 2 Freq.</th>
<th>All Extremes Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.05</td>
<td>0.107475053</td>
<td>0.066246678</td>
<td>0.125555556</td>
<td>0.092810458</td>
</tr>
<tr>
<td>1.25</td>
<td>0.059287167</td>
<td>0.033319427</td>
<td>0.066666667</td>
<td>0.062091503</td>
</tr>
<tr>
<td>1.36</td>
<td>0.043216183</td>
<td>0.022921467</td>
<td>0.022222222</td>
<td>0.013398693</td>
</tr>
<tr>
<td>1.44</td>
<td>0.04331691</td>
<td>0.021861555</td>
<td>0.022222222</td>
<td>0.012745098</td>
</tr>
<tr>
<td>1.52</td>
<td>0.087049512</td>
<td>0.040694446</td>
<td>0.042222222</td>
<td>0.049019608</td>
</tr>
<tr>
<td>1.68</td>
<td>0.043852041</td>
<td>0.018906041</td>
<td>0.022222222</td>
<td>0.022875817</td>
</tr>
</tbody>
</table>
From the two experiments that we have conducted, we can observe that our proposed ESBCH has determined the number of clusters dynamically. Also, our proposed iSPA has estimated correctly POT values such that the constructed GPD models have predicted precisely the probabilities of the fitted extremes.
CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

Delivering a reliable service offering to the mobile clients remains a challenging aspect in today’s telecommunications and computer networks. A broad number of research studies have undertaken the service evaluation process from one side; that is, the service-side perspective. Conversely, clients’ assessment to the service has been mostly neglected. In this research, we propose a client-side service monitoring and evaluation system which mainly relies on the clients’ assessment of the service offerings. Our proposed approach shows the strength of adopting the client-side service monitoring and assessment which flips the view of the service monitoring and evaluation operation from the service side to the client side.

The proposed approach consists of an overall architecture which combines powerful techniques and tools (such as EVT, LOP, and SNA) that are devoted to predict service degradation and network failures. Hence, the core of the proposed architecture is design to utilize the EVT model builder to converge to a precise QoS model which is used to fit the MCs’ collected network measurements (extremes) in order to assess the service behavior on the long-run. Also, an efficient management approach is presented through the BM to orchestrate between the MCs and service providers. The reliability of the constructed QoS models is also one of the main contributions of our research since the proposed system exploits the LOP technique to exclude unreliable measurements collected from un-trusted MCs so that it significantly improves the service evaluation process. Furthermore, the proposed architecture is developed to operate in both single and
multi-hop computer network environments. Also, it has the capability to deliver the required QoS guarantees to both cloud service providers and cloud service consumers by providing on-time service assessment.

In spite of the verity of network environments, it is important to mention here that the proposed system has been successfully deployed and tested on GENI infrastructure through the Seattle testbed.

The constructed experimental and analytical results prove the efficacies of the proposed client-side QoS system compared to traditional service management approaches. Furthermore, the outcomes demonstrate that our system, through its developed algorithms and approaches, improves the service performance by proactively predicing the service failures, enhances the service reliability through recognizing and excluding the data collected from un-trusted MCs. It also reduces the service monitoring and evaluation processing overhead since it deals only with selective data (extremes), and gives the service providers an alternative way to revise the SLA by considering clients’ perspective regarding the service offerings.

### 8.2 Research Contributions

We believe that applying the proposed client-based QoS approach provides immense advantages to the service evaluation process. Our proposed approach has created a big leap towards that direction since it investigates the possible techniques that can be utilized to make the mobile clients participate significantly in the service assessment process. Furthermore, it provides an adaptive improvement toward reducing the network failure by early detection and predication of the service degradation and generates consistent feedback to the service providers to improve the dedicated services;
thus, the services can be orchestrated efficiently. Our contributions can be summarized as follows:

1) The proposed client-based QoS approach characterizes the MCs perception of the service offerings through evaluating the collected network measurements without the need to insert measurement probes in the core transport network. Since the implementation of service monitoring and assessment tools is delegated to the MCs, our proposed architecture is more scalable and serves to offload the service from the monitoring burden inherent in traditional service management architectures. The proposed architecture can be implemented on different distributed environments such as cloud computing, GENI platform, and Hadoop/MapReduce framework.

2) Our proposed client-based QoS approach provides a unique utilization of EVT (GEV and GPD models). The EVT empowers the proposed architecture by adding efficient tools to model the collected extremes; thus, it interactively enables the proposed architecture to react to changes in the service behavior.

3) Toward converging to a reliable and precise service evaluation process, the proposed client-based QoS approach is designed to exclude out-of-profile data collected from MCs by exploiting the LOP, such that it provides the service providers with a better assessment tool to evaluate and improve their services. Moreover, our approach is based on algorithms capable of distinguishing and excluding the misbehaving MCs while maintaining or improving service performance prediction.
4) The proposed approach is capable of evaluating and managing the service offerings in both single and multi-hop network. Focusing on the multi-hop network management, the proposed approach utilizes the SLE and GEV techniques efficiently to predict network link performance degradation by estimating the delay extremes on each hop of a given network topology. Hence, service performance can be evaluated and improved through a continuous assessment process of the network’s links behavior over time.

5) Our client-based QoS approach is extended to overcome the limitations of the service evaluation process in the cloud computing environment. We present a novel client-based service monitoring and evaluation approach for cloud-based services and infrastructure. Toward that direction, the proposed approach has successfully converges to precise QoS model for early detection of the cloud service degradation. It proactively identifies cloud services and infrastructure with degraded performance by utilizing GPD model on the collected extremes from cloud mobile clients. Furthermore, it capable of providing multi-level service performance assessment through analyzing the collected extreme measurements from VMs, zones and datacenters. Then, we build precise QoS models to predict the performance of mobile clients that exhibit similar behavior by utilizing the strength of the SNA principles in conjunction with the EVT. Thus, we developed a novel SNA-based clustering algorithm that analyzes the strength of the interconnection links between Mobile Clients (MCs) and cluster related MCs in communities of similar behavior. The proposed approach is effective in providing
service providers with a better assessment tool to evaluate and improve their service offerings.

8.3 Future Work

In future research, our current results concerning the proposed client-based QoS approach can be extended as follows:

1) The client-based QoS management approach that utilizes the LOP can be applied in multi-hop networks. Such approach provides links level trust verification and performance assessment to the collected network measurements. In order to implement this approach, data must be collected from trusted MCs as well as reliable paths of the requested services since the service request can follow different paths based on the presented network topology. Thus, LOP can play an important role in choosing the most reliable path to deliver the service so that it can exclude unreliable links and its related data from the service evaluation process. Indeed, this approach will be effectively providing service providers with a better assessment tool to evaluate and improve their services by choosing only the trusted MCs and reliable routes.

2) Toward extend the reliability of the proposed multi-hop client-based QoS approach, we plan to measure the computational requirements of the algorithm and to what extent it can be implemented in real networks. This approach could also lead us to reduce the algorithm complexity such that the overall client-based QoS performance assessment process can be improved. An approach which improves network routing algorithms from the mobile clients’ perspective could also be implemented.
3) The characteristics of the MCs can be studied in depth to construct hierarchical clustered communities based on their underlying behaviors using SNA principles. Hence, either the agglomerative or divisive based clustering technique can be utilized depending on the theme of the collected data to explore MCs community structure. The constructed clusters can then be used to predict potential service degradations and failures in large-scale service deployments.

4) In order for it to cope with the emerging changes of the service-oriented Internet of Things (IoT) technologies[82], our proposed architecture could be extended to monitor and assess the QoS parameters of the IoT smart things/objects (i.e., network communication devices). The performance variation of the smart objects can be evaluated to capture potential degradations and failures as the nodes communicate amongst themselves and their environment. From this perspective, the proposed architecture (MCs and BM) interact with the smart objects in order to collect the network measurements and assess the objects’ behavior through building precise EVT-QoS model for a better IoT resource utilization.

5) Because our proposed client-based approach essentially involves in network resource monitoring and evaluation process, it is highly required to build visual tools that reflect the MCs perception of the offered service. Our proposed architecture could utilize the Random Art Approach (RAA)[83] to visualize the service extreme behavior; such utilization will be in terms of combining different EVT models using Backus-Naur Form (BNF) grammar rules to draw heat map like images. Thus, the generated random art will help the service provider to better understand the behavior of their service offerings.
BIBLIOGRAPHY


International Conference on Parallel, Distributed and Network-based Processing, pp. 120–124, Feb. 2012.


