NOVEL ANALYTICAL MODELLING-BASED SIMULATION OF WORM PROPAGATION IN UNSTRUCTURED PEER-TO-PEER NETWORKS

HANI SAYYAF ALHARBI

Doctor of Philosophy
Institute of Computing Science and Mathematics
University of Stirling

January 2017
DECLARATION

I, Hani Sayyaf Alharbi, hereby declare that the work in this thesis is original and has been composed by myself, except where reference is made to other works, and has not been submitted for examination for any other degree at this university or any other learning institutions.

Stirling, January 2017

________________________________

Hani Sayyaf Alharbi
ABSTRACT

Millions of users world-wide are sharing content using Peer-to-Peer (P2P) networks, such as Skype and Bit Torrent. While such new innovations undoubtedly bring benefits, there are nevertheless some associated threats. One of the main hazards is that P2P worms can penetrate the network, even from a single node and then spread rapidly. Understanding the propagation process of such worms has always been a challenge for researchers. Different techniques, such as simulations and analytical models, have been adopted in the literature. While simulations provide results for specific input parameter values, analytical models are rather more general and potentially cover the whole spectrum of given parameter values. Many attempts have been made to model the worm propagation process in P2P networks. However, the reported analytical models to-date have failed to cover the whole spectrum of all relevant parameters and have therefore resulted in high false-positives. This consequently affects the immunization and mitigation strategies that are adopted to cope with an outbreak of worms.

The first key contribution of this thesis is the development of a susceptible, exposed, infectious, and Recovered (SEIR) analytical model for the worm propagation process in a P2P network, taking into account different factors such as the configuration diversity of nodes, user behaviour and the infection time-lag. These factors have not been considered in an integrated form previously and have been either ignored or partially addressed in state-of-the-art analytical models. Our proposed SEIR analytical model holistically integrates, for the first time, these key factors in order to capture a more realistic representation of the whole worm propagation process.

The second key contribution is the extension of the proposed SEIR model to the mobile M-SEIR model by investigating and incorporating the role of node mobility, the size of the worm and the bandwidth of wireless links in the worm propagation process in mobile P2P networks. The model was designed to be flexible and applicable to both wired and wireless nodes.

The third contribution is the exploitation of a promising modelling paradigm, Agent-based Modelling (ABM), in the P2P worm modelling context. Specifically, to exploit the synergies between ABM and P2P, an integrated ABM-Based worm propagation model has been built and trialled in this research for the first time. The introduced model combines the implementation of common, complex P2P protocols, such as Gnutella and GIA, along with
the aforementioned analytical models. Moreover, a comparative evaluation between ABM and conventional modelling tools has been carried out, to demonstrate the key benefits of ease of real-time analysis and visualisation.

As a fourth contribution, the research was further extended by utilizing the proposed SEIR model to examine and evaluate a real-world data set on one of the most recent worms, namely, the Conficker worm. Verification of the model was achieved using ABM and conventional tools and by then comparing the results on the same data set with those derived from developed benchmark models.

Finally, the research concludes that the worm propagation process is to a great extent affected by different factors such as configuration diversity, user-behaviour, the infection time lag and the mobility of nodes. It was found that the infection propagation values derived from state-of-the-art mathematical models are hypothetical and do not actually reflect real-world values. In summary, our comparative research study has shown that infection propagation can be reduced due to the natural immunity against worms that can be provided by a holistic exploitation of the range of factors proposed in this work.
DEDICATION AND ACKNOWLEDGMENTS

In the name of ALLAH, the Most Gracious and the Most Merciful. I owe it all to him who is worthy of all praise for his blessings and granting me with the ability to undertake this research task.

**Dedication:** Throughout my life two people have always been there during those difficulties and trying times. I would like to dedicate this thesis and everything I do to my parents. In addition to them I have always been surrounded by their love and prayers. I would not be who I am today without the love and support of them.

**Acknowledgments:** Throughout my PhD journey, Professor Amir Hussain has been an excellent supervisor. He truly understands what hard work and dedication can bring to one’s life. I would like to express my deepest gratitude to him for his guidance and continues support. Without him, I could not have achieved this important milestone in my life. His insightful comments, constructive criticism and suggestions have given me invaluable guidance in various stages of my PhD. He has been very helpful and supportive throughout all these years.

I would like also to convey my sincerest gratitude to many people who shared their memories, experiences and support. A big thank you to Dr Jozsef Darkas, DhaifAllah, Omair, Sayed Nadeem and Kamran for their continuous encouragement and valuable suggestions and encouragement throughout my study.

Mostly, my sincere gratitude to the Ministry of Higher Education and Tabouk University in Saudi Arabia for granting me the scholarship and financial support to pursue my PhD at University of Stirling.

Lastly, my sincere appreciation for my life companions, my lovely wife and my kids, Malek Nawwaf and Farah (the pride and joy of our life). I am indebted to them for sharing every single moment with me and all their patience, support and continuous willing throughout the study.
LIST OF PUBLICATIONS


# CONTENTS

**CHAPTER 1: Introduction**

1.1. Background ................................................................. 19
1.2. Issue and Challenges ......................................................... 20
   1.2.1. Realistic Worm Propagation Models ................................. 20
   1.2.2. Mobility of Nodes ...................................................... 21
   1.2.3. User Behaviour Modelling ........................................... 21
   1.2.4. Real-World Dataset Availability ................................... 21
   1.2.5. Requirement for an Appropriate Modelling Tool .................. 21
1.3. Thesis Statement ............................................................. 22
1.4. Aims and Objectives .......................................................... 22
1.5. Contributions ................................................................. 23
1.6. Thesis Structure .............................................................. 24

**CHAPTER 2: REVIEW OF P2P NETWORKS**

2.1. Introduction ................................................................. 27
2.2. Distributed Systems .......................................................... 27
2.3. Peer-to-Peer Networks ....................................................... 28
   2.3.1. Introduction to P2P .................................................... 28
   2.3.2. Characteristics of P2P ............................................... 29
   2.3.3. Types and Classifications of P2P Systems ......................... 30
2.4. P2P Protocols ............................................................... 34
   2.4.1. Gnutella ................................................................. 34
   2.4.2. GIA .................................................................. 36
2.5. Mobile Peer-to-Peer Networks ............................................... 39
2.6. Issues and Challenges of P2P Networks ................................... 41
   2.6.1. Scalability Issue ....................................................... 41
   2.6.2. Free Rider Issue ....................................................... 41
   2.6.3. Node Mobility ......................................................... 41
   2.6.4. Social Networking .................................................... 42
   2.6.5. Conventional Modelling Tools ...................................... 42
   2.6.6. Security and Threats of Worms ................................... 42
2.7. Modelling Tools in P2P Networks .......................................................... 42
  2.7.1. Conventional Network-Modelling Tools ........................................ 43
  2.7.2. Oversim Simulator ........................................................................ 43
2.8. Agent-Based Modelling ...................................................................... 45
  2.8.1. ABM in P2P Networks ................................................................. 47
  2.8.2. Agent Based Modelling Tools ....................................................... 48
2.9. Comparison between ABM and Conventional Modelling Approaches .... 52
2.10. Evaluation of Unstructured P2P Networks using Conventional Simulation Approach ................................................................. 53
  2.10.1. Simulation Set-up ......................................................................... 53
  2.10.2. Performance Metrics ................................................................. 53
  2.10.3. Simulation Results ..................................................................... 54
2.11. Evaluation of Unstructured P2P Networks using ABM Approach ......... 56
  2.11.1. Model Description ...................................................................... 56
  2.11.2. Simulation Set-up ......................................................................... 59
  2.11.3. Performance Metrics ................................................................. 61
2.12. Comparative Analysis of ABM and Conventional Simulation Tools ....... 63
  2.12.1. Ease of Implementation .............................................................. 64
  2.12.2. Real-Time Analysis .................................................................... 65
  2.12.3. Design Evaluation ...................................................................... 65
  2.12.4. Visualization .............................................................................. 66
2.13. Summary .......................................................................................... 66

CHAPTER 3: SEIR: A NOVEL ACTIVE WORM PROPAGATION MODEL ........ 67

3.1. Introduction ......................................................................................... 67
3.2. Worm Propagation in P2P Networks .................................................. 67
  3.2.1. Code Red & Code Red II .............................................................. 68
  3.2.2. Nimda ......................................................................................... 68
  3.2.3. Slammer ..................................................................................... 69
  3.2.4. Blaster ....................................................................................... 69
  3.2.5. Conficker .................................................................................. 70
3.3. Modelling P2P Worms ...................................................................... 70
  3.3.1. Worm Propagation Models and Classification ............................... 70
  3.3.2. Immunization Strategies for P2P Worms ...................................... 78
3.3.3. Stochastic versus Deterministic Behaviour of Worm Propagation.................79
3.4. Factors Affecting the Worm Propagation Process........................................79
3.5. Requirements for a New Stochastic Analytical Model of P2P Worm Propagation.81
3.6. Proposed Discrete-Time Stochastic Active Worm Propagation Model (SEIR Model).................................................................82
  3.6.1. Notations ...............................................................................................84
  3.6.2. Derivation of Factors and Sub-Factors .....................................................85
  3.6.3. State Transitions and Factors .................................................................87
  3.6.4. Model Assumptions ...............................................................................88
  3.6.5. Relationship among Parameters .............................................................89
  3.6.6. Complete System of Equations ...............................................................90
  3.6.7. Boundary Conditions for SEIR Model ....................................................91
3.7. Conclusion.....................................................................................................92

CHAPTER 4 : SEIR MODEL: SIMULATION RESULTS AND DISCUSSION .......94
4.1. Introduction....................................................................................................94
  4.2.1. Infection Propagation over Time ..............................................................95
4.3. First Case Study: Comparative Evaluation of the Analytical Models Considering
  Gnutella Protocol ...............................................................................................98
  4.3.1. The Impact of Configuration Diversity on Infection Propagation...............99
  4.3.2. The Impact of User Behaviour on Infection Propagation .......................100
  4.3.3. The Impact of Random Values on Infection Propagation ......................100
  4.3.4. Random Values of User-behaviour for Homogeneous Network ............101
4.4. Second Case Study: Comparative Evaluation of Analytical Models Considering
  GIA Protocol ..................................................................................................102
  4.4.1. The Impact of Configuration Diversity on Infection Propagation ..........102
  4.4.2. The Impact of User Behaviour on Infection Propagation .......................103
  4.4.3. The Impact of Random Values on Infection Propagation ......................104
  4.4.4. Random User-Behaviour Values for Homogeneous Network ..............105
4.5. SEIR: The Impact of Different Parameter Values .........................................106
  4.5.1. The Impact of Configuration Diversity on Infection Propagation ..........106
  4.5.2. The Impact of User Behaviour on Infection Propagation .......................107
4.5.3. Random Parameter Values for Different Network Sizes .................................. 107
4.5.4. The Impact of Super-Peers on Infection Propagation .................................. 108
4.6. Discussion ........................................................................................................... 109
4.7. Performance Evaluation of SEIR Model using ABM Approach ....................... 112
   4.7.1. The Concept of Behaviour Space ................................................................. 112
   4.7.2. Description of SEIR Model based on ODD Protocol .................................. 112
   4.7.3. SEIR: Simulation Setup ............................................................................... 115
   4.8.1. The Impact of Configuration Diversity ......................................................... 117
   4.8.2. The Impact of User Behaviour ..................................................................... 120
   4.8.3. The Impact of Random Values of Parameters ............................................. 123
   4.9.1. The Impact of Configuration Diversity ......................................................... 125
   4.9.2. The Impact of User Behaviour ..................................................................... 127
   4.9.3. The Impact of Random Values of Parameters ............................................. 129
4.10. Analysis and Discussion ...................................................................................... 131
   4.10.1. Comparative Analysis of SEIR with Benchmark Models ......................... 132
   4.10.2. Comparative Evaluation of Conventional and ABM Modelling Approaches 134
   4.10.3. Variation of Different Parameter Values in SEIR Model ......................... 136
4.11. Validation of Theoretical and Practical Bounds of SEIR model ...................... 138
4.12. Summary ............................................................................................................ 140

CHAPTER 5: SEIR: REAL-WORLD EVALUATION ...................................................... 142
5.1. Introduction .......................................................................................................... 142
5.2. Conficker Spreading Mechanism and Behaviour ................................................ 142
5.3. An Analysis of the Conficker Worm Dataset ..................................................... 143
5.4. Related Work ....................................................................................................... 146
5.5. SEIR Model Evaluation with Conficker Worm ................................................... 148
   5.5.1. Performance Evaluation of SEIR Model ..................................................... 149
   5.5.2. Statistical Evaluation of Dataset ................................................................ 152
5.6. Discussion ............................................................................................................ 154
CHAPTER 6: A Novel M-SEIR Model: Investigating the Role of Mobility Factor....156

6.1. Introduction .............................................................................................................156
6.2. Related Work ...........................................................................................................157
6.3. Mobility Models in Networks ..................................................................................158
6.4. Factors Affecting the Worm Propagation Process in Wireless Networks ..........160
   6.4.1. The Gauss-Markov (GM) Mobility Model .......................................................160
   6.4.2. Bandwidth Factor ..............................................................................................161
   6.4.3. Infection Spread Probability by Mobile Node ..................................................162
6.5. Integration of Mobility Factor with SEIR Model ..................................................163
   6.5.1. Model Assumptions ..........................................................................................165
   6.5.2. Complete System ............................................................................................166
6.6. Evaluation of M-SEIR Model ..................................................................................167
   6.6.1. M-SEIR Model Evaluation using Conventional Tools ....................................167
   6.6.2. The Impact of Node Mobility Considering Gnutella Protocol .......................168
   6.6.3. The Impact of Node Mobility Considering GIA Protocol ...............................169
   6.6.4. Discussion .........................................................................................................170
6.7. Evaluation of M-SEIR Model using ABM Tools ...................................................172
   6.7.2. The Impact of Node Mobility on Worm Propagation Considering GIA ..........175
6.8. Mathematical Analysis of M-SEIR Model ............................................................176
   6.8.1. Configuration Diversity ....................................................................................176
   6.8.2. User Behaviour .................................................................................................177
   6.8.3. Node Mobility ..................................................................................................177
   6.8.4. Network Degree Distribution .........................................................................178
6.9. Summary ................................................................................................................178

CHAPTER 7: CONCLUSION AND FUTURE WORK .....................................................180

7.1. Introduction .............................................................................................................180
7.2. Thesis Summary .....................................................................................................180
7.3. Meeting the Objectives: Original Contributions ..................................................183
7.4. Limitations ..............................................................................................................185
7.5. Future Work ..........................................................................................................186
7. References..................................................................................................................195
LIST OF FIGURES

Figure 2.1: Classification of P2P Overlays [2] .......................................................... 31
Figure 2.2: Gnutella Search Algorithm ......................................................................... 35
Figure 2.3: GIA Pseudo-Code [16] ................................................................................ 37
Figure 2.4: Flowchart of GIA ......................................................................................... 39
Figure 2.5: Modular Architecture of OverSim [52] ......................................................... 45
Figure 2.6: NetLogo Main Interface ................................................................................. 50
Figure 2.7: Starlogo User Interface ................................................................................ 51
Figure 2.81: GIA Model in NetLogo ................................................................................. 60
Figure 2.92: Average Hop Count ..................................................................................... 62
Figure 2.103: Delay in GIA ........................................................................................... 62
Figure 2.114: Success Rate of Query in GIA ................................................................. 63
Figure 3.1: Simple SIR Epidemic Model [89] ................................................................. 71
Figure 3.2: SEIRS Model [94] ....................................................................................... 73
Figure 3.3: Classification of Worm Propagation Models and Addressed Parameters ....... 77
Figure 3.4: The States and Transitions .......................................................................... 83
Figure 3.5: Network Topologies ....................................................................................... 87
Figure 4.1: Infectious Population Density (STAWP model) ............................................. 96
Figure 4.2: Infectious Population Density (Four-Factor Model) ....................................... 97
Figure 4.3: Infectious Population over Time without Parameter Values ......................... 98
Figure 4.4: Configuration Diversity for all Models .......................................................... 99
Figure 4.5: User-Behaviour for all Models ..................................................................... 100
Figure 4.6: The Impact of Random Values of Parameters .............................................. 101
Figure 4.7: The Impact of Random User-Behaviour ....................................................... 102
Figure 4.8: Configuration Diversity for all Models .......................................................... 103
Figure 4.9: User-Behaviour for all Models ..................................................................... 104
Figure 4.10: Impact of Random Parameter Values ......................................................... 105
Figure 4.11: The Impact of Random Values for a Homogeneous Network ..........................105
Figure 4.12: Impact of Configuration Diversity over Time on Infection Density ...............106
Figure 4.13: Impact of User Behaviour over Time on Infection Propagation ..................107
Figure 4.14: Impact of Network Size on Worm Propagation (SEIR Model) .................108
Figure 4.15: The Impact of Super-Peer on the Worm Propagation Process .....................109
Figure 4.16: Life-Cycle of Worm for All States ..........................................................114
Figure 4.17: Netlogo GUI for Implementation ..................................................................117
Figure 4.18: Infection Propagation over Time for Gnutella with CD = 0.3 ......................118
Figure 4.19: Infection Propagation over Time for Gnutella with CD=0.5 .....................119
Figure 4.20: Infection Propagation over Time for Gnutella with UB=0.3 .....................121
Figure 4.21: Infection Propagation over Time for Gnutella with UB=0.5 .....................122
Figure 4.22: Infection Propagation with Random Values .............................................124
Figure 4.23: Infection Propagation over Time for GIA and CD=0.3 ............................126
Figure 4.24: Infection Propagation over Time for GIA, CD=0.5 ..............................127
Figure 4.25: Infection Propagation over Time for GIA and UB=0.3 ............................128
Figure 4.26: Infection Propagation over Time for GIA and UB=0.5 ............................129
Figure 4.27: Infection Propagation with Random Values .............................................130
Figure 4.28: Integrated SEIR Model..............................................................................131
Figure 4.29: Comparative Evaluation of ABM and Conventional Tools .......................135
Figure 4.30: Lower and Upper Bounds of SEIR model ..............................................140
Figure 5.1: Sample 1 from One Trace File of Conficker Worm ....................................145
Figure 5.2: Sample 2 from One Trace File of Conficker Worm ....................................145
Figure 5.3: Sample 3 from One Trace File of Conficker Worm ....................................146
Figure 5.4: Conficker Trends for Four Countries [125] ..............................................148
Figure 5.5: Infection Propagation in Hybrid Model [148] ...........................................149
Figure 5.6: Infection Propagation in VEISV Model with Different Reproduction Rates [156] .........................................................................................................................149
Figure 5.7: Infection Propagation considering Different Values ....................................151
Figure 5.8: Infection Propagation over Time ................................................................. 152
Figure 5.9: A Comparative Evaluation of SEIR with Real Data and Benchmark Models ... 154
Figure 6.1: Mobility Models in Wireless Networks [165] .............................................. 159
Figure 6.2: Infection Probability by Mobile Node ......................................................... 163
Figure 6.3: Heterogeneous P2P Network ................................................................. 166
Figure 6.4: Infection Propagation with Different Values considering Gnutella ............... 169
Figure 6.5: Infection Propagation with Different Values considering GIA ..................... 170
Figure 6.6: The impact of Variety Values on GIA and Gnutella .................................. 171
Figure 6.7: Impact of Node Mobility of Infection Propagation (M-SEIR Model) .......... 174
Figure 6.8: Impact of node mobility of infection propagation (SIM model) [164] ............ 174
Figure 6.9: The Impact of Node Mobility on Infection Propagation (PQ Model) [145] .... 175
Figure 6.10: Infection Propagation considering GIA Protocol (M-SEIR Model) .......... 176
LIST OF TABLES

Table 2.1: Unstructured P2P Overlays ................................................................. 33
Table 2.2: Conventional Modelling Tools .............................................................. 44
Table 2.3: GIA Default Values from (default.ini) .................................................. 54
Table 2.4: Division of Different Blocks of ODD Protocol ...................................... 57
Table 3.1: Notations and Description .................................................................... 85
Table 4.1: The Impact of Configuration Diversity on Different Models .............. 110
Table 4.2: Impact of User Behaviour on Different Models ...................................... 110
Table 4.3: The Impact of Random Values of Parameters on Different Models .... 111
Table 4.4: The Impact of Random Values on a Homogeneous Network ............. 111
Table 4.5: Die-out Behaviour for Configuration Diversity .................................... 133
Table 4.6: Die-out Behaviour for User-Behaviour .................................................. 134
Table 4.7: The Bounds of SEIR Model ................................................................. 139
Table 5.1: Maximum Infection Propagation by Conficker .................................... 144
Table 5.2: Sample 1 from One Trace File of Conficker Worm ............................. 144
Table 5.3: Range of Crobach’s Alpha ..................................................................... 152
Table 5.4: Dataset from SEIR model and Conficker Dataset ............................... 153
Table 5.5: Statistical Evaluation of SEIR and Conficker Dataset ....................... 153
Table 6.1: Notations and Description .................................................................... 165
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABM</td>
<td>Agent Based Modelling</td>
</tr>
<tr>
<td>AMM</td>
<td>Agent-Based Malware Modelling</td>
</tr>
<tr>
<td>BA</td>
<td>Barabasi Albert</td>
</tr>
<tr>
<td>BFS</td>
<td>Breadth First Search</td>
</tr>
<tr>
<td>BRW</td>
<td>Biased Random Walk</td>
</tr>
<tr>
<td>CAIDA</td>
<td>Centre for Applied Internet Data Analysis</td>
</tr>
<tr>
<td>CAIDA</td>
<td>Centre for Applied Internet Data Analysis</td>
</tr>
<tr>
<td>CD</td>
<td>Configuration Diversity</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma-Separated Values</td>
</tr>
<tr>
<td>DCOM</td>
<td>Distributed Component Object Model</td>
</tr>
<tr>
<td>DFS</td>
<td>Depth First Search</td>
</tr>
<tr>
<td>DFS</td>
<td>Distributed File Systems</td>
</tr>
<tr>
<td>DHT</td>
<td>Distributed Hash Table</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of Service</td>
</tr>
<tr>
<td>DSL</td>
<td>Digital Subscriber Line</td>
</tr>
<tr>
<td>ETLM</td>
<td>Extended Topology Logic Matrix</td>
</tr>
<tr>
<td>GES</td>
<td>Gnutella Efficient Search</td>
</tr>
<tr>
<td>GM</td>
<td>Gauss-Markov</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile communication</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>IDPS</td>
<td>Intrusion Detection and Prevention Systems</td>
</tr>
<tr>
<td>IIS</td>
<td>Internet Information Services</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>KBR</td>
<td>Key-Based Routing</td>
</tr>
<tr>
<td>KIT</td>
<td>Karlsruhe Institute of Technology</td>
</tr>
<tr>
<td>LCN</td>
<td>Least Common Neighbours</td>
</tr>
<tr>
<td>LSA</td>
<td>Local Search Approach</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LUNES</td>
<td>Large Unstructured Network Simulator</td>
</tr>
<tr>
<td>MAFDC</td>
<td>Mobility-aware file discovery control</td>
</tr>
<tr>
<td>MIT</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td>MSDE</td>
<td>Microsoft SQL Server Desktop Engine</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Address Translator</td>
</tr>
<tr>
<td>NFS</td>
<td>Network File System</td>
</tr>
<tr>
<td>ODD</td>
<td>Overview, Design Concepts, and Details</td>
</tr>
<tr>
<td>P2P</td>
<td>Peer-to-Peer</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Desktop Assistants</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PMP</td>
<td>Pontryagin’s Maximum Principle</td>
</tr>
<tr>
<td>PQ</td>
<td>Probabilistic Queuing</td>
</tr>
<tr>
<td>RPC</td>
<td>Remote Procedure Calls</td>
</tr>
<tr>
<td>RW</td>
<td>Random Walk</td>
</tr>
<tr>
<td>RWP</td>
<td>Random Way Point</td>
</tr>
<tr>
<td>SEIDR</td>
<td>Susceptible, Exposed, Infectious, Diagnosed, and Recovered</td>
</tr>
<tr>
<td>SEIR</td>
<td>Susceptible, Exposed, Infectious, Recovered</td>
</tr>
<tr>
<td>SFN</td>
<td>Scale Free Networks</td>
</tr>
<tr>
<td>SIM</td>
<td>Susceptible Infectious Mobile</td>
</tr>
<tr>
<td>SIR</td>
<td>Susceptible, Infected and Recovered</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>STAWP</td>
<td>Stochastic Active Worm Propagation</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TTL</td>
<td>Time to Live</td>
</tr>
<tr>
<td>UB</td>
<td>User Behaviour</td>
</tr>
<tr>
<td>VD</td>
<td>Virtual Degree Local search algorithm</td>
</tr>
<tr>
<td>XSL</td>
<td>Extensible Style sheet Language</td>
</tr>
</tbody>
</table>
CHAPTER 1 : INTRODUCTION

1.1. Background

Over the past decade, the internet has grown dramatically. With this increase, several applications have emerged to solve the various issues of scalability and resilience [1]. One of these applications is Peer-to-Peer (P2P) network. In keeping with the wide range of uses for P2P networks, a multitude of definitions for Peer-to-Peer have been proposed.

A P2P network has been defined as a distributed system comprising a set of interconnected nodes without the need for the intermediation or support of a centralized server [2]. Such a system is capable of adapting failure and accommodating scalability while also maintaining satisfactory connectivity and performance. This is done to share various resources such as content, storage and bandwidth. P2P networks are offered as overlay networks on the top of physical networks. The overlay network is responsible for location and storing services while the physical network is responsible for physically forwarding data.

P2P network paradigms have gained significant popularity due to their independence from the central server management. Since P2P networks are implemented as overlay networks, they allow a higher level of communication and data-sharing among participating peers. The connections that are established between peers are logical and independent from the physical layer [3]. A P2P network can be categorized in one of the two main categories which are structured and unstructured overlays.

The popularity of P2P networks has led to serious security implications and vulnerabilities [4]. Common threats to P2P networks are sabotaging the network through internet worms, which can disable a large portion of a network [5][6][7], and compromise privacy and identity, resulting in the disclosure of users’ private information [8], fairness in sharing [9], and the distributed denial of service attack, making the servers unavailable to requesting clients [10], [11], free riders [12], and even the blocking of P2P traffic.

Among all the security-related challenges, P2P worm remains one of the major vulnerabilities. A P2P worm is a kind of malicious code that spreads by means of P2P networks [13]. The network worm is a self-reproducing program that, unlike a virus, does not infect other...
programs. Instead, a worm creates copies of itself which then further creates even more copies [14].

Worms target multi-processing operating system, create copies of itself and spread to other connected neighbours on network. There are many reasons that make P2P networks vulnerable to worms. One of the reasons is that many P2P networks, such as Kaaza [15], Gnutella [16] and GIA [16], are able to share an executable file. Another reason is that an enormous number of common users are connected to P2P network at any given time. These users are normally not aware of the security concerns related to P2P networks that can potentially make the network vulnerable to a potential attack [17].

1.2. Issue and Challenges

The focus of this thesis is worm propagation modelling in unstructured P2P networks. There are several primary reasons why these particular issues have been addressed. Primarily, worm propagation modelling is achieved by developing analytical models that take different sets of parameters into consideration. One of the common concerns of some of these models is when only a subset of parameters is considered while assuming constant values for major factors, which may act as a hard-constraint for the analytical models. Moreover, the worm propagation process is a complex process and the simulations generated using conventional tools raise questions concerning the validity of the results obtained. Another important consideration is the separate handling of P2P protocols and worm propagation models. Since some P2P worms are designed for a particular P2P protocol, an integrated solution reflecting more accurate behaviour is required.

The issues and challenges addressed in this research are as follows:

1.2.1. Realistic Worm Propagation Models

A number of analytical models were developed to show the worm propagation process. These models were based on different factors such as configuration diversity, mobility, the join-and-leave behaviour of nodes and the infection time-lag. Most of the models follow a deterministic approach [6] while a few are modelled as stochastic [7]. The issue with these particular models is that these were developed using a subset of causative factors instead of integrating all the relevant factors and therefore, as a result, they fail to portray a more realistic worm
propagation process. Consequently, the immunization strategies developed using these models suffer from a high false-positive rate for the detection of worms.

1.2.2. Mobility of Nodes

In presence of diverse sets of devices, such as hand-held personal desktop assistants (PDA), smart phones, laptop computers and desktop machines that are connected through a P2P network, it is essential to consider the impact of the mobility of the node on not only on query propagation but also on worm propagation [18]. Relatively less attention has been paid in the literature to consider the mobility factor in analytical models.

1.2.3. User Behaviour Modelling

This is a broad term which includes the online/offline behaviour, the setting-up upload/download limit, the free-rider issue and many more behavioural issues. During the worm propagation process, different P2P clients log on and off frequently, set up upload and download limits randomly and utilize the network as free-riders. Therefore, determining the impact of these factors on worm propagation is an important issue. These factors need to be addressed in the development of more realistic analytical models.

1.2.4. Real-World Dataset Availability

The availability of a real-world dataset showing the infectious node and the propagation behaviour is a real challenge in understanding the worm process. Many of the existing research is based on hypothetical assumptions and associated worm propagation models [19] and relies on simulations to model the behaviour, and, therefore, has not been validated through data generated by real networks.

1.2.5. Requirement for an Appropriate Modelling Tool

Considering the complexity and adaptive nature of the worm propagation process, the conventional network modelling tools are limited when it comes to modelling an exact reflection of the process. To address this issue, a promising modelling paradigm, Agent-based Modelling (ABM), in P2P worm modelling context has been used. Specifically, to exploit the synergies between ABM and P2P, an integrated ABM-Based worm propagation model has been built and trialled in this research for the first time.
1.3. Thesis Statement

P2P networks suffer from many security issues, one of them being worm propagation in a highly connected network where all peers are sharing their data. This thesis investigates the propagation process of worms affected by different factors such as configuration diversity, infection time lag and user-behaviour. In this thesis, an integrated analytical model is developed to demonstrate the influence of these factors, holistically, using a P2P protocol. The propagation of active P2P worms is also simulated on P2P protocols (Gnutella and GIA), using an analytical model with different parameter values.

Furthermore, the mobility of nodes is addressed within the context of P2P networks and the mobility factor is computed using existing mobility models and wireless-network-specific parameters. Moreover, using conventional and ABM tools, extensive simulations are comparatively evaluated, and these not only reflect the effectiveness of the analytical model itself but also investigate the feasibility of ABM tools in the domain of worm propagation modelling. Moreover, the proposed analytical model is further evaluated using a real world data set of Conficker worms by considering other benchmark models that have been inspired and developed, based on the same data set.

1.4. Aims and Objectives

The key objective of this thesis is to develop a comprehensive analytical model for the propagation of worms. The aim is to discover the impact of different factors on the worm propagation process and to understand the more realistic foot-prints of worms in an unstructured P2P network. This understanding will contribute to the adoption of counter-measurement strategies against the worms and to a reduction in the false-positive detection ratio exhibited by the current detection models. The objectives of this thesis are the following:

- To investigate different P2P networks, conventional network modelling tools and the feasibility of agent-based modelling tools in simulating the worms in P2P networks, as well as to analyse comparatively both types of tools, considering different metrics such as query-delay and hop-count.
• To investigate existing worm propagation models in P2P networks, and classify models according to the worm propagation mode that has been addressed. Also, to investigate the parameters affecting the worm propagation process.

• To represent a more realistic picture of worm propagation by developing an analytical model for worm propagation that holistically integrates the important factors. Moreover, to evaluate the worm propagation using the proposed model and considering infection density as the major metric. The evaluation will make use of both types of modelling tools (Conventional and ABM).

• To evaluate the proposed model with a real-world worm dataset along with benchmark models developed on top of the same dataset. Also, to discuss the role of different network-related parameters, such as the Network Address Translation (NAT) protocol, on infection density.

• To investigate the role of node mobility on the process of worm propagation. This is achieved by developing the mobility factor for mobile nodes using mobility models. This is followed by the integration of the mobility factor with the existing analytical model for worm propagation. To evaluate the proposed analytical model using the various tools and to demonstrate the node importance of node mobility on the worm propagation process.

1.5. Contributions

In this thesis, a number of issues related to P2P worms has been investigated and addressed. These include: (1) the development of a more realistic worm propagation model in unstructured P2P networks, combining different factors; (2) the impact of mobility on the worm propagation process; (3) the evaluation of the proposed model considering a real-world dataset of Conficker worm; (4) worm propagation modelling issues in conventional network modelling tools; and (5) the exploitation of Agent-Based Modelling (ABM) in a P2P domain with the secondary objective of implementing an integrated model of P2P protocol (Gnutella or GIA) with the worm propagation process.

The expected contributions of this thesis are the following:

• The development of an analytical model for the worm propagation process in a P2P network, considering different factors such as the configuration diversity of nodes, user
behaviour and infection time lag. The factors are generally not, or at least only, partially considered in most of the analytical models. In the proposed SEIR model, they will be integrated holistically in order to provide a more realistic representation of the actual effects of the worm propagation behaviour.

- An investigation into the effect of the mobility of nodes, the size of worm and the bandwidth of wireless links on the worm propagation process in mobile P2P networks. The model is designed to be flexible as both wired and wireless nodes are considered. There is also the flexibility to calculate the mobility factor for each individual node at any given times provided instead of computing for the whole network.

- As a secondary objective, a new and promising modelling paradigm, namely, Agent-based Modelling (ABM), is explored. This is found to be an effective and powerful tool across a wide range of domains. In this thesis, the capability of such an ABM modelling approach in simulating P2P network and P2P worm propagation has been demonstrated. Consequently, for the first time, an integrated model, implementing the unstructured P2P protocols (Gnutella and GIA) along with the aforementioned analytical models, has been carried out. Moreover, a comparative study of ABM versus conventional network modelling tools is also undertaken.

- An exhaustive and comparative evaluation of the worm propagation process, using the proposed model with conventional and ABM tools. A discussion of the associated issues and challenges encountered during the simulations. In addition, a discussion on various artefacts which are produced as a result of these simulations.

- An evaluation of the proposed model considering the data set of one of the most recent worms (Conficker). The verification of this model is achieved using ABM and conventional tools compared with benchmark models developed and based on the same data set.

1.6. Thesis Structure

This thesis consists of seven chapters, a synopsis of which is given below.

- **Chapter 1** gives an introduction to the thesis. It presents the thesis statement, the challenges, the objectives and the contributions of the research.
• **Chapter 2** introduces P2P networks, their characteristics and types, namely, unstructured and structured P2P systems. It provides further details concerning P2P protocols such as GIA and Gnutella, the issues and challenges associated with P2P networks and conventional modelling tools. Moreover, Agent-Based Modelling in P2P networks is introduced and this is followed by a comparative discussion between ABM and conventional simulators. A performance evaluation of GIA using a conventional simulator (Oversim) and an ABM tool (Netlogo) is carried out. This chapter lays the foundation for integrating the implementation of P2P protocols along with the analytical models of worm propagation, using ABM tools.

• **Chapter 3** provides a comprehensive review of P2P worms and their classification, their propagation behaviour, and their destruction strength and of existing analytical models by presenting the current state-of-the-art of the existing worm propagation models. Following this, gaps in the existing research are highlighted and parameters, such as the infection time-lag, configuration diversity and user behaviour are considered as important parameters in the worm propagation process. More importantly, an entirely new SEIR model is presented, one that holistically takes all of these important parameters into account.

• **Chapter 4** gives exhaustive results for SEIR model, which unveiled in Chapter 3. The results were generated using both ABM and conventional tools. A detailed and comparative analysis for worm propagation behaviour according to the different values of the factors is generated. In addition, an integrated solution for the proposed model and P2P protocols (Gnutella and GIA) is implemented. The impact of different tools (ABM and Conventional) is evaluated from the perspective of the worm propagation process.

• **Chapter 5** evaluates the proposed SEIR model with a real data set from one of the most recent worms (Conficker). The results were obtained using conventional and ABM tools and then compared with the real data in the presence of selected benchmark models.

• **Chapter 6** extends SEIR model to M-SEIR model by investigating the role of node mobility, the bandwidth of wireless links and the worm data size as important factors and then computes the node mobility factor, based on the Gauss-Markov mobility
model. The throughput of wireless links is also analytically computed. The additional factors related to mobility of nodes are integrated with SEIR analytical model. Without losing generality of the network, M-SEIR model can be applied on wired, wireless or heterogeneous networks. The salient features of this model are that it can maintain the spatial-temporal behaviour of nodes, the node-level mobility-factor control, the randomness control for individual nodes, and the throughput offered by the network and the worm size.

- **Chapter 7** concludes the thesis. It provides an overall summary of the whole thesis, revisits the thesis objectives and discusses how these have been addressed. Finally, the limitations of this study are discussed and potential directions for future research are suggested.
CHAPTER 2: REVIEW OF P2P NETWORKS

2.1. Introduction

This chapter provides an overview of P2P networks, their characteristics and taxonomy. It further describes important P2P protocols and their functions. Existing issues and challenges in P2P networks are then briefly discussed. Following this, an overview of conventional network modelling tools is provided. Moreover, the utility of Agent-Based Modelling (ABM) tools is also introduced within the domain of P2P networks by considering a case study implementation. Moreover, comparative evaluation between ABM and conventional network modelling tools is given by implementing GIA protocol using OverSim and Netlogo simulators. The obtained results of this implementation indicate certain benefits from ABM tools such as ease of implementation, real-time analysis support being an aid to visualization.

2.2. Distributed Systems

The distributed systems are basically designed to share data with each other and form the basis of many distributed applications [20]. Also, it allow multiple processes to share data over long periods in a secure and reliable way. These processes serve as a basic layer for distributed systems and applications. Additionally, different types of architectures followed in distributed file system, and a brief description of these architectures is provided as follows:

1- **Client-Server Architecture**: The basic theme of client-server architecture is to share data among different clients in standardized format. To achieve this objective, Network File System (NFS) was proposed which represents file independent of the files system on local clients. NFS is a standard which can be utilized for sharing files by any application program without relying on local file system and provide remote access to files by the application [21].

2- **Cluster-Based Distributed File Systems**: Parallel applications that require parallel computing are mainly hosted on multiple cluster servers. The associated file systems of the cluster will be adjusted accordingly. A file to be stored on distributed servers is
divided into multiple parts and these parts are stored on the different servers thus making parallel processing possible [22].

3- *Symmetric Architectures*: The typical examples of full symmetric organizations are based on P2P technology like Chord DHT-based system. These architectures are centralized which are based on distributed hash tables (DHTs) [23] and decentralized such as given in [24].

In following sections, the distributed systems will be discussed in detail in the context of P2P networks.

### 2.3. Peer-to-Peer Networks

#### 2.3.1. Introduction to P2P

The internet has recently grown dramatically. With this increase, several applications have emerged to solve various issues of scalability and resilience. One of these applications is P2P. In accordance with the wide range of use for P2P, many definitions for Peer-to-Peer systems have been proposed. P2P system is decentralized system of connected nodes without any central mechanism of mediation [2]. This system is capable of adapting failure and accommodating scalability while maintaining satisfactory connectivity and performance. This is done to share various resources such as content, storage and bandwidth. P2P systems can mainly be divided into three different architectures, namely, centralized, decentralized structured and decentralized unstructured [2]. In a centralized approach, central servers are dedicated to manage the operations of P2P network. In contrast, decentralized P2P architecture eliminates the need for such servers. This in turn provides flexibility among peers to exchange information between each other. In decentralized structured systems, nodes are distributed according to the distributed hash table (DHT), which shapes the whole topology into a specific structure as, for example, in Pastry, Kadameila, and Chord. In a decentralized unstructured system, there is neither a dedicated server nor any distributed hash function. The most popular examples are Gnutella [16] and KaZaA [25]. There is a further common distinction in that P2P can be classified as structured, unstructured overlays.
2.3.2. Characteristics of P2P

Some of the significant characteristics are considered as principles for Peer-to-Peer system paradigm and make it useful in today’s technology can be summarized as follows:

- **Resource-sharing**: To support the operation of Peer-to-Peer system, each peer contributes some resources such as network bandwidth, disk storage and compute cycles.

- **Decentralization and Symmetry**: In P2P system, there are no dedicated servers. Moreover, every node has a similar role and responsibility in the overlay and the collection of their behaviour shapes the overlay. However, some P2P algorithms give some special peers special roles such as super peers and relay peers.

- **Self-Organization**: The overlay is not controlled and dominated by a specific peer. The overlay organization is based on the local state and information at each peer. All peers cooperate in the overlay maintenance and formation. The network also tends to improve its state throughout the run of the network.

- **Networked**: All the peers can reach each other in the same overlay. Since all peers are interconnected with each other in one graph, the peers in the network will participate in forwarding and propagating messages in the network.

- **Stability**: Even with a high churn rate (the rate of peers joining and leaving the network), P2P system is designed to be stable and maintain connectivity throughout the graph by recovering from failure and reshaping the graph, based on changes in the topology, as can be seen in Fault-Tolerance.

- **Autonomy**: In P2P system, the capability of each peer is determined according to its own resources. Each peer is also able to independently determine whether to join or leave the overlay and what request it should send to the overlay.

- **Scalability**: Due to the autonomy factor, Peer-to-Peer systems need to be highly scalable. This is critical as the number of peers can change dramatically and unpredictably. Currently, several P2P applications are operating with a significant number of peers.

- **Fault-Tolerance**: Since the behaviour of the peers is not predictable, P2P systems are designed to be tolerant. In case, a peer goes offline or is disconnected from the overlay,
its resources will be reallocated. This is possible since some degree of redundancy is built into the system.

2.3.3. Types and Classifications of P2P Systems

The numerous designs for P2P networks have led to many proposals for classification [2]. One of the common distinctions is to categorize P2P overlays into structured and unstructured overlays. Structured overlays are distinguished by a variety of dimensions such as the maximum number of hops, the routing algorithm, or the size of overlay with node degree and lookup type. Moreover, structured overlays use key-based routing to facilitate object placement, where the nodes maintain a routing table about how to reach the entire overlay nodes. Furthermore, unlike unstructured overlays, they limit the number of messages needed to discover any specific object in the overlay. Examples of structured overlays are Chord, Kademlia and Pastry. Contrasting with structured overlays, unstructured overlays are usually further distinguished by how search requests propagate, by the distribution of node degree in the peer population, and by differences in link creation with neighbouring nodes. Nodes in unstructured overlays depend only on their adjacent nodes for propagating messages and do not impose any particular network structure. Most commonly, message propagation strategies use to some extent flooding and random walk approaches [26]. Most deployed P2P applications employ unstructured search techniques [27]. More recently, research has focused on the design of efficient search algorithms, including query propagation and query processing.

Various proposals for the classification of P2P systems are based on the different design of P2P networks. One design involves completely decentralized architecture while other design combined servers with P2P routing.

The classification of P2P overlays is shown in Figure 2.1 below:
2.3.3.1. **Structured Overlay**

A structured P2P network is a network in which nodes maintain routing information about reaching all other nodes in the network cooperatively [28]. Structured overlays set a limit on the number of messages that can be used to propagate a query in the network [29]. As compared with unstructured overlays that do not provide any limit on query propagation in the network. This becomes important particularly while searching low priority queries [16].

Local routing information is maintained by each node which is used by the forwarding algorithm. A specific bootstrap procedure is utilized to initialize the routing table when a peer joins or leaves the overlay. The routing table information is periodically exchanged by the peers as part of overlay maintenance.

The structured P2P overlay can use key-based routing in which each address is associated with a set of keys in the address space so that the nearest peer to an address stores the values of the associated keys. Since address space is virtualized and the addresses of peers are randomly assigned, the topological neighbours of the peer are distant in the underlying overlay network. Although it provides fault tolerance, it nevertheless has a significant performance overhead. As a result, the topology-aware overlays used measurement of proximity of peers in the underlying network to create neighbour peer in the network [30].
2.3.3.2. Unstructured Overlay

In contrast to structured overlays, unstructured overlays do not impose any specific structure on the overlay itself. Unstructured overlays are usually classified by how the search requests propagate, by the distribution of node degree in the peer population, and by the differences in link creation with neighbouring nodes. Nodes in unstructured overlays depend only on their adjacent nodes for the propagating of messages [27]. The lack of structure poses many challenges especially when searching the network. To solve this issue, many search approaches [31] have been introduced and can be classified as follows:

- **Blind Search Methods**: Is used in overlays in which the peers do not maintain any information about the location of resources.
- **Informed Search Methods**: In contrast to the blind search technique, informed search methods require peers to maintain further routing information about other peers in the network in order to forward queries to the appropriate peer.

The search is the most fundamental function of most deployed P2P overlays, and most of the studies of unstructured P2P tend to enhance the search process and design it more efficiently.

An unstructured network is a network in which a peer relies on its neighbouring peer for the delivery of message to any other peer in the network. Example of message propagation strategies are Flooding (broadcasting the search query to its immediate neighbours) [1] and Random walk (forwarding the query to a randomly selected neighbor at each step until the query is satisfied with sufficient responses) [31]. A description of the message propagation strategies is provided in Section 2.3.1 in the context of Gnutella protocol.

The unstructured network graph can be compared with power-law or scale free random graphs [32] where random graphs are the graphs showing small world phenomena and social networks. The most common P2P overlay network designs are listed in Table 2.1:
Table 2.1: Unstructured P2P Overlays

<table>
<thead>
<tr>
<th>Type</th>
<th>Design</th>
<th>Features</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hill climbing backtracking</td>
<td>Freenet</td>
<td>Routing and security can be provided using hill climbing with backing</td>
<td>[33]</td>
</tr>
<tr>
<td>Hill climbing backtracking</td>
<td>Fast Freenet</td>
<td>Each peer creates its own summary and share it with other peers in Fast Freenet.</td>
<td>[33]</td>
</tr>
<tr>
<td>Hill climbing backtracking</td>
<td>Small World Freenet</td>
<td>Free net considering a small network</td>
<td>[34]</td>
</tr>
<tr>
<td>Random Walk</td>
<td>GIA</td>
<td>Based on four amendments and techniques such as active flow control, dynamic topology adaptation, biased random walk and one-hop index replication to improve performance</td>
<td>[16]</td>
</tr>
<tr>
<td>Random Walk</td>
<td>LMS</td>
<td>Consistent hashing of object identifier to place objects near a close node identifier</td>
<td>[35]</td>
</tr>
<tr>
<td>Flooding</td>
<td>Gnutella</td>
<td>Flooding of requests by super peers on behalf of regular peers</td>
<td>[36]</td>
</tr>
<tr>
<td>Flooding</td>
<td>FastTrack</td>
<td>Connection shuffling with super peer architecture being a proprietary protocol</td>
<td>[37]</td>
</tr>
</tbody>
</table>

2.3.3.3. Hybrid Overlays

Hybrid overlays refer to a mixture of purely decentralized such as Gnutella protocol and centralized such as Kaaza protocol. A hybrid approach is followed by Gnutella protocol 0.6 [38] by combining centralized and decentralized models. The nodes of overlay are categorized as leaves and ultra-peers. An ultra-peer keeps connection to fewer leave nodes and connected to other ultra-peers. These peers work as proxy for query directing to leave nodes with the believe that the leave nodes will response the query.

2.3.3.4. Hierarchical Overlays

Hierarchical overlay networks can be represented structured P2P networks look up service [39]. The service is based on distributed hash table (DHT) functionality. The data is search based on keys which are distributed on multiple serving nodes. In contrast hierarchical DHTs, in which peers are organized into groups, and each group has its autonomous intra-group
overlay network and lookup service. The top level peer in each group takes the responsibility for a key and determines the specific peer which is responsible for the key.

2.4. P2P Protocols

P2P protocol works at the application layer of P2P networks. The major responsibility of these protocols is to form and maintain the overlay network and the participating nodes. It provides a node with the mechanism of searching for contents, joining the network, leaving the network and publishing it contents. Starting from 1999, there is a long list of P2P protocols. A few of them are Ares, Bitcoin [40], BitTorrent[41], FastTrack [42], eDonkey [42], Gnutella and GIA. In the next section, popular P2P protocols will be discussed in more detail.

2.4.1. Gnutella

The idea of P2P file sharing system was devised by Napster while supporting centralized servers for file sharing. Soon after that, the idea of decentralized file sharing and search algorithm was proposed in the form of Gnutella which is an open, decentralized and search protocol, mainly used for file sharing. The term ‘Gnutella’ refers to the overlay network of Gnutella-speaking applications connected via the internet and a number of smaller and often private disconnected networks. Gnutella gained popularity after Napster was closed down and it has turned out to be one of the most popular systems to date [43].

Primarily, using Gnutella protocol, when a new peer joins the system, it executes a bootstrapping process to find and connect to potential peers. Once a new peer has connected to the overlay, it sends a ping message to all connected peers to announce its presence. The receiving peer sends back a pong message which in turn comprises its port number and IP address.

Query propagation in Gnutella is based on a flooding mechanism, for example the search query propagates from the original requesting peer to all connected neighbours. The query is further flooded until it finds the contents or time-to-live expires. Moreover, the intermediate nodes are in charge of checking their repository as to whether they have the desired file, and of replicating and forwarding the query to all reachable neighbours in the overlay. The query-hit message or query response is also returned along the reverse path in the direction of the requesting peer. Gnutella protocol follows random walk procedure in which the peer which
have to forward query to its immediate neighbours selects its neighbour randomly without any criteria.

To reduce the potentially enormous number of messages in the network, queries are restricted by a time-to-live (TTL) value, which signifies the number of hops a message can travel before it is discarded. TTL is decreased by one with each hop, until the message is discarded when it reaches zero. Broadcasting the query to all reachable peer nodes in the overlay within the TTL value limitation is called a Breadth-First Search (BFS) mechanism [44].

![Gnutella Search Algorithm](image)

Figure 2.2: Gnutella Search Algorithm

Figure 2.2 shows the search process of Gnutella. Assume that peer ‘A’ needs to find a particular file. Initially, it generates its search query which is forwarded to all its connected neighbours, peers B, C, and E. When those neighbours receive the query message, they will check whether their repository has the required file. Otherwise, they forward the query on to their neighbours. Suppose that peer D holds the required file. Peer D will send a response along the reverse path to the peer that sent the query, which was peer C. After that, peer C continues forwarding the response to the query originator, peer A. Eventually, peer ‘A’ contacts peer D directly to download the required file.

The design goals of Gnutella are as follows:
• **The Ability to Operate in a Dynamic Environment:** Gnutella based application operates in an environment where the nodes join and leave the networks very quickly. In order to provide the flexibility for smooth operations, transparent resource handling is required.

• **Performance and Scalability:** P2P network can only be beneficial for large-scale networks where the limitations of client-server paradigm become evident. The scalability refers to the capability of a network to handle a large number of the participants.

• **Reliability:** The network attacks should not significantly degrade the performance of network significantly.

• **Anonymity:** It is about how to protect the privacy of users seeking or providing unpopular information.

2.4.2. GIA

One of the influential unstructured P2P approaches is GIA [16], which derives its name from GIAnduia, and is based on the common P2P Gnutella algorithm. By 2007, Gnutella, which GIA is improving, was the most popular file-sharing protocol with an estimated market share of more than 40%. As Gnutella adopted a decentralized search algorithm, it had one fundamental problem, namely, that the nodes became overloaded quickly due to the effect of the high aggregation query rate [16]. Therefore, as the resource placement performance is significantly affected by the overlay topology, GIA suggested several modifications to Gnutella design, to deal with the high aggregation query rate. These can be summarized as the following four components:

• **An Active Flow Control:** To ensure that the nodes are not overloaded with high queries, each sender node can direct the query only to a neighbour that has informed the sender by means of a flow-control token. Each token represents a one-query message that the neighbour is ready to accept.

• **One-hop Replication:** Each node keeps an index for its connected neighbour’s content. So, when a node receives a query message, it responds with the matching contents, and it will also include a list of contents of its connected neighbours.
• **A Biased Random Walk (BRW) Search Protocol**: Instead of using flooding or random-walk search methods, the search process in GIA makes sure that all search queries are directed to the high-capacity nodes rather than to randomly chosen nodes as in random walkers. The algorithm for the BRW is shown below in Figure 2.3.

• **A Dynamic Topology Adaptation**: The most important component in GIA ensures that all nodes in the overlay are connected to the high-capacity nodes, based on the pseudo-code in Figure 2.3. To achieve the main goal of topology adaptation, each node in GIA calculates a value from 0 to 1. This value is presented by Level of Satisfaction (S), in order to measure how the node is satisfied with its connected neighbours. The value of 0 means that the node is completely unsatisfied whereas 1 refers to that the node being totally satisfied.

```
Let $C_i$ represent capacity of node $i$
if $num_{nbrs}_X + 1 \leq max_{nbrs}$ then {we have room}
   ACCEPT $Y$; return
{we need to drop a neighbor}
subset ← $i \forall i \in nbrs_X$ such that $C_i \leq C_Y$
if no such neighbors exist then
   REJECT $Y$; return
candidate $Z$ ← highest-degree neighbor from subset
if ($C_Y > max(C_i \forall i \in nbrs_X)$) {Y has higher capacity}
or ($num_{nbrs}_Z > num_{nbrs}_Y + H$) {Y has fewer nbrs}
then
   DROP $Z$; ACCEPT $Y$
else
   REJECT $Y$
```

**Figure 2.3: GIA Pseudo-Code [16]**

Where:

- $Num_{nbrs}$: Number of neighbours of a peer.
- $Max_{nbrs}$: Maximum number of neighbours that a node can have.
- $C_i$: Capacity of a node to handle requests.
With the aim of improving the consistency of GIA overlay networks in mobile networks, the study in [45] presented a new technique based on GIA. The main concept introduced by M-GIA is the physical distance between two nodes. The information about a node’s location will therefore be considered as an additional parameter to GIA protocol. Indeed, the distance between nodes does not comply with the mobility of nodes. Thus, this in turn will burden the network, thus increasing the need to calculate the distance with each movement. Furthermore, some studies such as GES [46] have inspired the idea of topology adaptation in GIA to improve their look-up performance, with the difference bring that the topology adaptation in GES is mainly used to organize the semantically relevant nodes into similar semantic groups.
2.5. Mobile Peer-to-Peer Networks

Most recently, the way people interact and share content with others and connect to their environment has been affected by the significant growth in communication and wireless technologies, particularly with mobile phones.
In general, mobile phones are characterized by four features, which make a distinction between P2P and other conventional wired devices and which affect the interaction with the overlay. The four features are as follows:

- **Roaming**: Where mobile devices may move from one mobile network to another mobile network and, in turn, their IP addresses could be consequently changed.

- **Node Heterogeneity**: The network capacity is not equally distributed among all nodes. Due to the physical position of nodes, one node can have more capacity compared with another node; hence it can work as a super peer node while the other node still works as an ordinary node.

- **Energy Limitation**: Power consumption for all mobile devices is a critical issue, which must be taken into account. Different mobile hardware manufacturers are now providing power-saving modes to save the limited energy of their devices.

- **Multi-Homed**: Nodes are capable of establishing connections with different networks at the same time.

Recently, wireless nodes represented significant percentage of all overlay nodes in the form of smart phones, PDA and other wireless devices. These wireless devices are themselves well-integrated into our modern way of life.

The suitability of mobile phones for applications that depend on unstructured P2P for information exchange nevertheless presents a number of challenges. Utilizing existing mobile applications, mobile nodes not only support content delivery but also act as a data provider. Some of the challenges of deploying P2P over mobile networks are listed below:

- **Energy Constraints**: Unfortunately, battery capacity is not keeping pace with the huge advances in mobile phone and wireless technology. This must be taken into account for any future developments.

- **Query Propagation**: Considered one of the critical issues for data-sharing in P2P overlays, the plain flooding mechanism is quite a simple solution for this issue in wired networks. It nevertheless seems challenging to guarantee it in bandwidth-limited networks. Further refined schemes will undoubtedly be required.
• **NAT and Firewall Traversal:** This was created by mobile operators in order to allow only outbound connection. Semi-centralized servers or proxies are proposed as a solution to the aforementioned problem [47].

• **Bandwidth Constraints:** P2P was introduced for adoption on wired networks that depend on IP infrastructure and feature bandwidth availability. Moreover, wireless connections remain limited in bandwidth.

• **Security:** The popularity of P2P networks has led to security implications and vulnerabilities [33]. Common threats to P2P networks, such as worms, privacy and identity, fairness in sharing, the distributed denial of service attack, and the blocking of P2P traffic, can all sabotage the network.

### 2.6. Issues and Challenges of P2P Networks

#### 2.6.1. Scalability Issue

Scalability of a system refers to the capability of a system to operate under varying loads without significant drop in the performance [48]. The access upload bandwidth is probably the main limitation of P2P content distribution because most of the links have asymmetric and low upload capacity [48]. As a result, most of P2P protocols have to set a relatively low access bandwidth which results in performance bottleneck. The example of protocols suffering from the scalability issue is given in Gnutella, Tapestry, Pastry and Chord.

#### 2.6.2. Free Rider Issue

P2P networks work on the resource-sharing principle of resource, where each node contributes to the system by offering its own resources and, in return, expect some other nodes to provide it with the required resources. The nodes that take the required resources from other nodes but are reluctant to share their own resources are called ‘free riders’ [49]. As a result, no node is willing to offer its contents and expects other nodes to bear the cost instead. Ultimately, this results in resource congestion and unavailability.

#### 2.6.3. Node Mobility

With growing advancements in wireless technologies, more and more users are shifting to mobile devices [50]. For a P2P network that consists of wired and wireless nodes, it is vital to
devote significant research efforts to model query propagation behaviour in wireless networks, particularly when the impact of node mobility on the overall process is considered.

2.6.4. Social Networking

The social media apps are taking over the business and majority of community is using these apps [51]. In the paradigm of P2P networks, as these apps are increasing communication, it is important that research is done on the behaviour of users during usage of these applications.

2.6.5. Conventional Modelling Tools

The behaviour of P2P networks is simulated using existing conventional tools. However, the query or worm propagation process in P2P network is a complex process that requires modelling support from tools to handle the behaviour effectively. Moreover, particular aspects of P2P simulations such as visualization of the propagation process and the die-out behaviour of simulations are not considered by the existing modelling tools.

2.6.6. Security and Threats of Worms

A P2P worm is a software program with a malicious code that treats the network as a medium of spreading. There are many reasons that make P2P networks vulnerable from worms. One major reason is that many P2P networks, such as Kaaza, Fastrack and eDonkey, are able to share an executable file. The second reason is that at any given time a huge number of common users are connected to P2P network. These users are normally unaware of the security concerns related to P2P networks.

Since P2P worms cause significant damage in terms of financial losses, a detailed study about worm types, propagation modes, the efforts towards understanding the behaviour of worms, and immunization strategies is provided in Chapter 3. The impact of different factors on the propagation process and the integration of these factors into one analytical model has been achieved.

2.7. Modelling Tools in P2P Networks

Network modelling has been done using a variety of tools provided by open-source community and proprietary vendors. While all of these tools have advantages and
disadvantages, certain tools nevertheless fail to model protocol-specific behaviour. In this context, P2P networks have complex query propagation and protocol specific challenges. A list of conventional tools handling these challenges, along with brief description is provided in following sections.

2.7.1. Conventional Network-Modelling Tools

Different modelling tools are used in networks to model a variety of behaviours. An overview of the current tools and frameworks used for the existing P2P networks is given in Table 2.2, where the main advantages and drawbacks of using each of them are also noted.

2.7.2. Oversim Simulator

OverSim [52] is an overlay network simulation framework, developed in C++ at the Karlsruhe Institute of Technology (KIT) and based upon OMNeT++, in order to handle the underlay network. OverSim includes a number of P2P models for structured ones such as Kademila, Chord, and Pastry, and unstructured ones such as GIA. It has been designed to be modular comprising three levels, as shown in Figure 2.5:

- **Application**: Where the tested applications are defined, such as the look-up method key-based routing (KBR), and direct hash table (DHT).
- **Overlay**: This comprises a number of structured and unstructured models.
- **Underlay**: Different kinds of exchangeable models underlying a network, which comprises three network models, namely, Simple, SingleHost and INET. These help to simulate the heterogeneous and large-scale networks.
Table 2.2: Conventional Modelling Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
<th>Programming language</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMNeT++ [53]</td>
<td>Highly modular discrete event simulator to evaluate networking systems</td>
<td>C++</td>
<td>- Graphical User Interface (GUI) and high-level description language (NED).</td>
<td>- The number of available protocols is not as high as alternative simulators.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Run distributed simulations through multi- machines.</td>
<td>- Requires high programming skills to do modification.</td>
</tr>
<tr>
<td>NS-2 [54]</td>
<td>Discrete-event simulator targeting network protocols with packet-level accuracy.</td>
<td>C++, OTcl</td>
<td>Large community of users</td>
<td>Little work reported in P2P overlay.</td>
</tr>
<tr>
<td>OverSim [52]</td>
<td>Framework for simulating P2P overlays, built upon OMNeT++</td>
<td>C++</td>
<td>Easy to: Evaluate the robustness and resilience of structured P2P overlays under realistic different churns.</td>
<td>Not supported for unstructured P2P overlays.</td>
</tr>
<tr>
<td>PeerSim [55]</td>
<td>Focuses on P2P protocols, supports two simulation models, cycle- and event-based</td>
<td>Java</td>
<td>Simulate very large number of cycle-based nodes</td>
<td>Avoid the details of underlying network - Low accuracy</td>
</tr>
<tr>
<td>P2PSim</td>
<td>Multi-threaded discrete event simulator for structured overlays</td>
<td>C++</td>
<td>Support structured P2P with churn behaviour</td>
<td>No GUI and visualization tool for the simulation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Little support for extending P2P protocols</td>
</tr>
<tr>
<td>Overlay Weaver [56]</td>
<td>Toolkit for simulating P2P network, focuses only on structured overlays</td>
<td>Java</td>
<td>Easy to use to test new protocols</td>
<td>Real-time execution, and thus it is difficult to replicate results or control parameters during experiments. Limited use and barely documented</td>
</tr>
<tr>
<td>PlanetSim</td>
<td>The architecture comprises 3 layers: network, overlay, and application</td>
<td>Java</td>
<td>High scalability with a large scale and a big number of nodes.</td>
<td>Implements just basic underlay models. Does not provide a visualization during simulation or GUI.</td>
</tr>
</tbody>
</table>
OverSim users need to define their scenario parameters in the omnetpp.ini script. While the default script default.ini contains the default values and parameters of each P2P overlay, there is no need to define any parameters in the onmetpp.ini file if the user is going to use the default values.

While observing OverSim simulator, it has been found that OverSim provide controls over simulation rather at abstract level by setting up configuration parameters. Also, there is a minimal control on P2P protocols related parameters. Moreover, defining the behaviour of query propagation through analytical models is hard to do and evaluate.

2.8. Agent-Based Modelling

Agent-Based Modelling (ABM) is a type of computational modelling. This model is for simulating the actions and interactions of autonomous agents [57]. These agents can be either individual or collective entities such as organizations or groups. In this section, an introduction to Agent-Based Modelling will be provided. Also, the differences between ABM approaches
and classical approaches will be discovered and discussed. ABM can be used to develop agents which are proactive, autonomous and intelligent. The pro-activeness refers to agents’ capability of taking the initiative, enabling them to communicate with other agents and make decisions on their own [57].

Most of the classical simulators are not designed to deal with a model in which the number of events can exponentially increase, which makes the model inefficient and hard to be analysed [58]. Agent-based modelling provides a description for the complex systems in P2P paradigm with a higher level of abstraction [59]. This can provide a better understanding of P2P paradigm and can be used for further improvements.

While there is no unified definition of what an agent is, there are nevertheless three common characteristics for the agents:

- **Autonomy**: Agents can function without any kind of incorporation or intervention by other programs or users.
- **Reactivity**: Agents can react to the changes in the environment according to their goal.
- **Social Ability**: Agents in the system will interact with each other and possibly with the users. These interactions are done with the intention of enhancing cooperation and competition.

Depending on the field of study, some characteristics have been presented in the literature [60], these are as follows:

- **Pro-Activity**: Agents are capable of initiating actions as well as reacting to changes in the environment. These actions are done in order to achieve a particular goal.
- **Adaptation/Learning**: Agents in the system can improve by learning and adapting to its environment.
- **Mobility**: Agents are able to move between nodes in a network.

As mentioned earlier, a peer could autonomously decide when to join or leave the overlay. Reactivity is noticeable in P2P paradigm where peers are expected to react to their environment. By reacting to their environment, peers try to make the network converge to an optimal state. The peers in the network are proactive. They will therefore take advantage of opportunities to improve their situation within the network. Moreover, adaptation and mobility
will allow nodes to learn and improve their performance as they traverse through the network, discovering links and resources.

### 2.8.1. ABM in P2P Networks

With the intention of improving search efficiency in P2P network, a number of studies have suggested different search methods to resolve this issue. Moreover, many agent-based technologies have been proposed to improve the search efficiency in unstructured P2P. For example, a study by [61] proposed the Least Common Neighbours’ local search approach (LCN). LCN has been compared with the Virtual Degree Local search algorithm VD, which is based on selecting nodes with a high virtual degree to transmit the queries (the number of the node’s neighbours that have not received queries). One of the goals of LCN approach is to expand the scope of the search within the network, by propagating the queries from the node that is already occupied by the agent to a node that has the least common neighbours. LCN approach also attempts to decrease the possibility of repeatedly forwarding and receiving query messages. However, these two approaches will in fact produce too much redundancy.

In [62], the authors consider merging two paradigms, P2P and ABM. This paper also dealt with how to improve the existing P2P ideas and concepts by using ABM. After reviewing the two paradigms, a new prototype system was suggested called ‘BestPeer’. The architecture of BestPeer comprises three layers: an agent layer at the top of the hierarchy and an underlay P2P layer, to support low-level communication, file-sharing and network configuration. This system assumes that the whole system is not known by any peer. However, the system will use resource-sharing and discovery concepts in P2P to apply it in ABM. ABM is applied on a P2P grid computing paradigm in [63] in a discussion about how the tasks in grid computing are handled and distributed using an agent-based model.

In [64], the paper discusses the use of an ant-based approach to route queries in peer-to-peer systems. This paper also discusses the parameters that affect the performance of the system. One parameter, in particular, is the ratio of agents or ants that are exploiting the current best path over the agents that are exploring and searching for a better path.

Ecological mechanisms have inspired some researchers to develop ideas to solve P2P problems. For example, in studies in [65][66], researchers peer-selected a random number of
neighbouring peers to send queries to, and a set of micro-scale rules controlled the search mechanism. The agents had to consume the associated energy with peers in order to survive. A study by [67] discussed the major threat of the propagation of a P2P worm in the internet, how it poses a serious challenge to the network security, and how it becomes increasingly complex. It worth mentioning that most of the existing studies resolve this problem by focusing only on the topology of the network as the main parameter affecting the worm propagation function. This study, however, introduces an additional effect, namely, the time taken by the worm to affect the peer. The results of this study were collected by simulating this proposal and developing new agent-based and analytical models.

All of these studies show the applicability of ABM to the domain of P2P networks. In subsequent sections, different ABM tools will be discussed in order to model the complex propagation behaviours in P2P networks.

2.8.2. Agent Based Modelling Tools

Over the last few years, several agent-based simulators have appeared and been discussed in general in many subjects, such as biology, physics and social studies. Recently, these studies have also covered some P2P aspects. This section reviews a number of the existing tools for network simulation that are focused on either conventional protocols or on some agent-based and biologically inspired protocols.

Several agent-based simulators have been dedicated to the explicit discussion of some aspects of P2P, such as biologically inspired simulators. Examples of these simulators and frameworks are Anthill [68], LUNES [69], OverSwarm [70] and Netlogo [71]. Anthill framework was introduced specially to support the design of ant-based algorithms in P2P and the implementation and evaluation of these algorithms. Moreover, the execution of algorithms in Anthill could be cycle-based, enabling large-scale simulations with a large number of nodes, or based on a real-world set-up. The focus of Anthill development was on the framework rather than on the actual algorithm.

Gabriela [69] presented an agent-based Large Unstructured Network Simulator (LUNES) to simulate the large number of nodes in complex networks. LUNES was introduced as a modular approach, establishing the three separate stages of network creation, protocol simulation and
result evaluation. The reason underlying the notion of this approach was to facilitate the integration and the reuse of some existing software tools. This module also allows analysis of a result with the use of integrated tools based on a simple template file.

Another promising platform, called OverSwarm, was proposed by [70] to facilitate the evaluation and implementation of P2P protocols, based on bio-inspired paradigms. The main feature of the OverSwarm platform is its integration with the popular P2P simulator (OverSim).

A few popular ABM tools have been discussed in the following sections from the perspective of their utilization in P2P domain.

2.8.2.1. **NetLogo**

NetLogo is an agent-based framework based on the Logo language and it is principally more suitable for modelling complex systems that have evolved over a period of time [59]. The NetLogo model has been used in several domains such as the social, biological, economic, chemical and psychological sciences as well as in numerous other natural sciences [57]. This has inspired some researchers to simulate and analyze distributed systems such as mobile ad hoc networks [58] and modern wireless networks [72].

A NetLogo package comes with an integrated code library containing numerous numbers of examples, and with a well-documented code to help the user easily modify and build their simulation. There is such a variety of examples covering so many subjects that most of the time it is easy to find a similar example to what the user is seeking to solve.

The layout of the NetLogo interface, as shown in Figure 2.6, contains several components:

- A Virtual World Screen made up from patches and where the turtles and links are set.
- A collection of sliders, buttons, inputs and monitors to control the creation and interaction of turtles.
- Turtles: the important component in the Logo world, which is interacting with the patches in the Virtual World.
- Patch: represented by a single place where the turtle exists.
- Observer: to observe everything throughout the virtual world and ask the turtles to distribute and interact over the patches.
These turtles are defined with specific variables such as direction or position. The user is also able to locate turtles in a Virtual World interface at any position and initiate interaction between these turtles according to the code implementation that defines their behaviour. After that, the outcome of this interaction between turtles can be collected and presented in real-time with monitors and charts in the NetLogo graphic interface (See Figure 2.6).

As it can be seen in Figure 2.6 above, the layout of NetLogo is formed by the virtual world graphical window where the nodes are interacting together, with different sizes according to its capacity. The sliders in the upper-left are also used to define the number of nodes as well as the minimum and maximum number of links. On the right side of Figure 2.6, there are three monitors for the hop-count. Moreover, there are also some buttons for creating the searchers, changing the layout and running the simulation. A few of the important features of Netlogo are given below [71]:

- A simple language structure, based on the logo language.
- A huge number of agents and variables.
- A comprehensive dictionary for all NetLogo language primitives.
- An extensive Models Library with numerous documented code examples.
- A friendly and multi-purpose Interface.
A collection of monitors and sliders to inspect and control the interaction between agents.

2.8.2.2. StarLogo

Start Logo was developed by MIT and adopted by academia in countries world-wide. The objective of Star Logo is to provide a playful, cooperative environment while also ensuring an adequate structure for learning of how to build models. The graphical user interface of Star Logo is shown below in Figure 2.7:

![StarLogo User Interface](image)

Figure 2.7: Starlogo User Interface

2.8.2.3. Repast Symphony

Repast is an abbreviation of REcursive Porous Agent Simulation Toolkit and was developed as a pure java implementation; it is now used widely for agent-based modelling and simulations. The features of Repast S are a visual aid for ABM design, a behaviour specification of agents, multiple executions of the model and the ability to carry out a comprehensive examination. The basic objective for the development of Repast S was to visually specify the properties of models. Since the framework is embedded with an Eclipse
Integrated Development Environment (IDE\(^1\)), there is no separate installation package required for this tool.

2.9. Comparison between ABM and Conventional Modelling Approaches

The main difference between agent-based modelling and a conventional network modelling is the nature of the agents, which could be autonomous, proactive and intelligent. As was mentioned earlier, ABM agents are able to take the initiative by contacting other agents and making decisions on their own, whereas the objects (Classes) in conventional tools are simple and reactive and have a limited capability. Furthermore, the ABM can be distinguished from

---

\(^1\) http://www.eclipse.org/

\(^2\) https://repast.github.io/images/relogo_execution.png
the conventional network simulators by its simplicity, flexibility and efficiency in its various types of modelling systems [72].

Moreover, one of the main factors that attracted the attention of users and increased the popularity of using ABM is the capability to visualize and animate the dynamic interaction between agents, which is also considered to be a powerful means of verifying, validating and understanding such models. Over time, the reputation of ABM has increased as it is capable of managing the increase in complexity of real-world systems, consisting of an enormous number of interacting agents that are autonomous and adaptive.

2.10. Evaluation of Unstructured P2P Networks using Conventional Simulation Approach

In this section, the performance of an important unstructured P2P protocol, Gnutella-like (GIA), is evaluated using an OverSim simulator. The value of different metrics such as latency and hop count are measured and evaluated. The simulation is done on a Windows platform.

2.10.1. Simulation Set-up

Before initiating this experiment in OverSim, users need to define their scenario parameters in the omnetpp.ini script. While the default script default.ini comprises the default values and parameters of each P2P overlay, there is no need to define any parameters in the omnetpp.ini file if the user is going to use the default values. OverSim is based upon the OMNeT++ framework, in order to handle the underlay network, whereas the results are processed by OverSim. This experiment was carried out by varying the number of network sizes i.e. 500, 1000, 1500, 2000, 2500 and 3000 nodes; the underlying reason behind limiting the size of nodes and the ability to simulate a larger network was the current limitation of computer power. The results of the simulation are stored in the results file, produced by OverSim and processed in Microsoft Excel.

2.10.2. Performance Metrics

The following performance metrics were evaluated from the conducted simulation:

- **Hop Count**: The mean number of hops between the source node that initiates the query and the requested node that has the desired file.
• **Latency:** Measures the duration of the time, in seconds, needed for the stabilization process after facing churn, as well as the duration of time needed for resolving a search query from creating until responding.

### Table 2.3: GIA Default Values from (default.ini)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The maximum number of neighbours (maxNeighbors)</td>
<td>50</td>
</tr>
<tr>
<td>The minimum number of neighbours (minNeighbors)</td>
<td>10</td>
</tr>
<tr>
<td>Topology adaptation interval (maxTopAdaptionInterval)</td>
<td>120 seconds</td>
</tr>
<tr>
<td>The delay between two update messages (updateDelay)</td>
<td>60 seconds</td>
</tr>
<tr>
<td>The time before message regarded to be lost</td>
<td>180 seconds</td>
</tr>
<tr>
<td>The time before neighbour regarded to be lost</td>
<td>250 seconds</td>
</tr>
<tr>
<td>Interval of issuing tokens (tokenWaitTime)</td>
<td>5 seconds</td>
</tr>
<tr>
<td>The time to send new key list to neighbours (keyListDelay)</td>
<td>100 seconds</td>
</tr>
</tbody>
</table>

### 2.10.3. Simulation Results

In Figure 2.9, the latency is computed by increasing the number of nodes in the network. Each of the nodes is generating queries at an instant of time. The capacity of the node is considered independent of the query generation rate. It can be observed that the delay starts increasing with the increase in number of nodes.

![Latency in GIA](image)
In Figure 2.10, the hop count is observed by increasing the size of the network from 500 to 3000 nodes. It can be seen that the hop count increases as the number of nodes in the network increases. This increase in hop count is due to the increase in the number of queries per node, which causes additional delay as well as hop count.

![Figure 2.10: Average Hop Count in GIA](image)

Different performance evaluations have been carried out for small and large networks ranging from 500 to 3000 nodes respectively. The evaluations have been carried out with the life-time of a node considered to be 300 seconds. This is done to measure how GIA would behave under a node’s churn. The scalability of GIA has been tested by measuring the hop-count against the increase in network size. It was observed that the value increases significantly as the size increases. This was also predicted by [16]. It was also noticed that the latency increases as the number of nodes and hence the number of queries increase with time. The latency increases as the query rate increases from individual nodes; this means that if the network size increases, the latency also increases.

The results verify the work done in the benchmark paper [16] for latency and hop count. The effect of churn rate and topology adaptation on bandwidth consumption cannot be verified using OverSim due to limited support for the configuration settings.
2.11. Evaluation of Unstructured P2P Networks using ABM Approach

In this section, a Gnutella-like (GIA) protocol is implemented using Netlogo (an ABM tool). The metrics to be studied are one-hop replication, delay in query propagation and bandwidth allocation. The simulation is verified from the ODD (Overview, Design Concepts, and Details) standard, as described below.

2.11.1. Model Description

ABM tools have been widely accepted in many domains. However, unlike mathematical models which are easier to read and understand, there is no standard method for describing the simulation models, which are difficult to reproduce and to understand. ODD is a protocol to describe different aspects of the simulation model [73]. The protocol consists of three components (Overview, Design Concepts, and Details) and is further divided into seven elements. These elements are Purpose, State Variables and Scales, Process Overview and Scheduling, Design Concepts, Initialization, Input and Sub-Models. A division different blocks is shown below in Table 2.4:
For Netlogo simulation and result generation, ODD protocol is followed. A description of the proposed simulation model according to ODD protocol is provided as follows:

### 2.11.1. Purpose

The object of the query propagation model is to study different matrices in a defined interval of time. The matrices to be studied are the latency, the success rate and the hop-count for GIA protocol under different query loads.

### 2.11.2. State Variables and Scales

The model consists of three levels: individual, network and environment. Individuals are nodes that are operated by a user and set with different factors such as capacity, minimum neighbours, maximum neighbours, replication rate, and token for flow control. The values of these factors are set via the GUI of the model.
At the network level, the degree distribution, the number of links, the network being with or without super-peers and being a scale-free network (SFN) are the properties of the network.

The environment refers to the setting of the variables related to visual display and processes. The variables in the model are network-type, layout-type, search behaviour and start-node-number.

2.11.1.3. Process Overview and Scheduling

The model proceeds in discrete time-steps. With each time-step, the query propagates and searches the contents. The query-originator searches for the most connected immediate neighbour and requests for a token to send the request. If the super-peer node is not occupied, it sends a token to serve the request. If the super-peer node is occupied and does not have the capacity to serve, the originator node has to wait until the node becomes free. If the super-peer has the contents, it responds to the originator node with a pointer to the contents or else it forwards the request to its most connected neighbour. The flow control of GIA model is explained in Figure 2.3.

2.11.1.4. Design Concepts

- **Emergence**: The query is initiated by a single node but the propagation is dependent on many factors. These factors are protocol-dependent.
- **Adaptation**: The nodes request the super-peer for contents and if the super-peer has the capacity to serve, it provides a token and the requester moves forward. If the searcher node finds the contents of the query, it is a successful search or else it is a failed search.
- **Fitness**: The nodes should be connected in the network. If any node gets disconnected due to any reason, it no longer participates in the query propagation process.
- **Prediction**: The probability of selection of the most suitable neighbour for the query propagation. This probability is dependent on GIA protocol-related factors.
- **Sensing**: The node must receive the query request in order to transfer the request to its immediate neighbours. The process is defined by using GIA protocol.
- **Interaction**: Nodes communicate with each other using GIA protocol.
- **Stochasticity**: There is no stochastic behaviour involved in the query propagation. The query’s propagation is based on the determined process of the protocol.

- **Collectives**: There is no grouping of individual agents. All nodes are spatially independent and purely distributed and connected using P2P-scale-free networks.

- **Observation**: To observe the query propagation, two methods of search take place, breadth-first and depth-first are observed. It is important to know that GIA follows breadth in its first search.

### 2.11.1.5. Initialization

The model can be initialized for one searcher node. The node could be any node selected from a randomly distributed network of nodes. The number of nodes can be specified from the GUI of the model.

### 2.11.1.6. Input

In general, the total number of nodes in the network, the minimum number of neighbours of a node, the maximum number of neighbours of a node, the search type, the node capacity and the starter node are provided as input to generate the scale-free network of nodes.

### 2.11.1.7. Sub-Models

The sub-models utilized by ABM in order to make the main model work perfectly.

#### 2.11.2. Simulation Set-up

In this section, the implementation of GIA protocol using a NetLogo simulator is discussed. Different parameters were observed and a comparison was made based on the benchmark work done by [16].

The parameters used to model GIA protocol are described below:

- **Size of The Network**: Due to the limited processing capability of the machine available, it was decided to model a network of 100 and 1000 nodes.

- **Query Rate**: This is the rate at which searches are initiated. Different query rates are taken under different network sizes and other parameter values.
- **Replication Rate**: This value represents the percentage of nodes that can satisfy the request. The range is from 0 to 1 where zero represents no replication and one represents 100% of the nodes having their replicated nodes.

- **Minimum Number of Links**: This is the minimum number of neighbours that a node can have.

- **Maximum Number of Links**: This is the maximum number of neighbours that a node can have.

- **Time-to-Live (TTL)**: This is the time a node experiences for getting a response

Netlogo provides a good level of understanding of P2P model. The current implementation of GIA protocol in Netlogo has produced and demonstrated it to be a useful educational tool. It is suitable for learners, being able to show them how a query is propagated through the nodes in the overlay, according to the predefined behaviour for the search method. The model also includes a real-time analysis, providing a good understanding of the communication between nodes. Moreover, the model visually represents the effect of the capacity on the nodes; the visual size of the nodes in this model represents its capacity; for example, bigger nodes have more capacity and vice-versa (Figure 2.10 below).

![GIA Model in NetLogo](image)

**Figure 2.81**: GIA Model in NetLogo

In (Figure 2.11), 1000 nodes are simulated in the virtual world; each node in the model has a different size representing its capacity. Each node is also connected to a number of neighbours,
based on the topology adaptation feature of GIA [16]. The red node can be clearly seen; it represents the target node of the searchers. With the input button, any number of researches can be created, with a single search being initiated each time the button is pressed. Based on the flow-control specification on GIA, the interaction between nodes will not begin until tokens are available for its connected neighbours, informing them that they are willing to accept their query. The upper-left button, named the ‘setup-network’, is used to create a network with predefined number of nodes.

The graph in Figure 2.10 illustrates the interaction for 1000 nodes with a Time-to-Live (TTL) of 50, and it can be seen that the monitors on the right side illustrate the average hop-count and the success/fail ratios. Other monitors on the left side show the number of success and fail searches. This explanation of GIA model, implemented in NetLogo, shows that it is indeed a tool well-suited for the purpose of demonstrating search methods and their effectiveness on the network.

2.11.3. Performance Metrics

The following metrics were collected as a result of the simulation:

- **Average Hop Count**: The average number of hops between the source node that initiates the query and the requested node that has the desired file.
- **Average Delay**: The duration time or query from start to finish.
- **Success Ratio**: The number of queries that successfully found the desired source.

From Figure 2.12, it can be seen that the average hop count was small for the small network size but it increased significantly as the network size increased. The average hop count for a network of 100 nodes was 3 hops, while the hop count for a network of 1000 nodes was 5 hops. Also, the replication rate was assumed to be 0.1%.
From Figure 2.13, it can be seen that the average delay increased smoothly with the increase in query rate. However, a sudden increase in delay was observed as the number of queries rose above 15000 queries. This experiment was performed with 1000 nodes where each node generated equal traffic. Interestingly, the same pattern was shown in [16].
With a 0.01% replication rate, it can be seen that the success rate of queries dropped significantly (Figure 2.14). This is due to the limited capacity of nodes to handle the requests and therefore result in time-out even if the resource is present in the network.

![Success rate graph](image)

**Figure 2.14: Success Rate of Query in GIA**

2.12. **Comparative Analysis of ABM and Conventional Simulation Tools**

GIA protocol was implemented in OverSim simulator and with Netlogo tool. The machine on which the results were generated was core2duo with 6 gigabyte (GB) of memory capacity. During the implementation and result generation, the following critical differences were found:

1- The run-time of the simulation, even for a smaller number of nodes, was too long in the Oversim simulator, while the run-time of the model execution and the results generation in Netlogo was significantly less. The reason for this is the number of parallel threads that Oversim created and could not handle due to the buggy code.

2- The Oversim simulator crashed after a long execution time for simulation of more than 3000 node network. During the execution process, a linear increase in memory over time was observed. This phenomenon reveal memory leak problem in the code and hence the program crashed. Since Netlogo is based on the Java programming language [76], the memory management is significantly better, compared with Oversim.

3- Oversim provides a configuration setting with limited options of different variable settings. External libraries and packages are required to convincingly simulate the
protocol behaviour. It requires a significant learning curve while, at the same time, not much support is available for the configuration setting for the simulator. On the other hand, Netlogo not only provides the parameter setting using different GUI controls but also a programming option in Logo programming language. Different built-in models increase the support at the next level.

4- ABM tools (including Netlogo) are general-purpose tools to solve problems of complex adaptive systems [77]. These tools, however, are not explicitly designed to simulate networking-related processes. For example, in a network simulation, 5000 to 10,000 nodes are normally required. Netlogo can support a limited number of nodes due to the implicit constraints imposed by the visualization of the process at the same time.

5- The ‘behaviour space’ tool in Netlogo is provided in order to understand the overall behaviour of the process under different parameter combinations, such procedure is not possible manually. The same support to understand the behaviour of a model is not provided in Oversim. The ‘behaviour space’ module of Netlogo, however, provides promising support to repeat the simulation virtually any number of times and to record the results in common formats such as XSL, CSV or simply a spreadsheet. The extraction of results thus becomes uncomplicated, as compared with a conventional tool such as Oversim, which records the results in a log file. These results can then be extracted using a certain extension, which itself requires significant configuration.

6- The description of any process with respect to time is handled in customized units ‘ticks’ instead of standard units such as ‘seconds’. The ‘tick’ parameter could have varying units of time and the process cannot be explained in terms of standard units. This is a major weakness of Netlogo.

In the next section, the findings and differences between the two different types of simulators for modelling P2P networks will be discussed. The advantage of using Netlogo for studying P2P models will also be covered.

2.12.1. Ease of Implementation

Compared to conventional simulators, it is noticeable that the time needed in Netlogo to build up the protocol or the program is significantly less. The Netlogo interface is well-suited to be
a useful tool for understanding how nodes interact according to their predefined behaviour. Furthermore, the source code for several example models, covering different scenarios and subjects, is fully available in the models library. These models can be found in the integrated library with the Netlogo package. These examples enable a new model to be easily produced simply by modifying existing models.

2.12.2. Real-Time Analysis

The OverSim simulator represents a promising tool for P2P overlays, especially for structured P2P networks. The simulation results are stored in the result files (scalar and vector files). There is, moreover, no easy way to visualize and analyse the result with a large number of nodes. In order to do so, the user needs either to add graphical user interface (GUI) scripts to the main implementation or to export the results file to the integrated development environment (IDE) of OMNeT++ tool. While the process of analysing the data is straightforward in NetLogo, the results are automatically generated in real-time while the simulation is running. The results can be visualized using real-time monitors and graphs on the user interface panel.

2.12.3. Design Evaluation

Since classical simulators such as OverSim require a sizable chunk of the system memory to implement the TCP/IP stack, the run-time for some scenarios is very long to get the results. However, since ABM relies heavily on abstraction, the run-time for comparable processes is significantly shorter. Moreover, the number of simulated nodes in unstructured overlays is limited in OverSim to fewer than 4000 nodes. Although OverSim is built over an INET framework, the platform is not very up-to-date as it depends on an old version of INET, thus preventing the user from using the new features in the new release of INET and OMNeT++. OverSim is also not a completely open source project, and this limits the users from learning the core of this framework.

Netlogo not only enables the active participation of users in all design, implementation and evaluation/prototyping of unstructured P2P overlays but also facilitates, through HubNet, user-learning through participatory simulations where users can, for example, in a future case
study, act as super nodes in order to learn about various aspects of the complex GIA protocol, such as network topology adaptation, flow control and search mechanisms.

2.12.4. Visualization

One of the underlying reasons for using NetLogo is that it comes with a friendly graphical user interface, comprising a dynamic communication between nodes, a collection of buttons and sliders to facilitate control of the variable inputs and automatic real-time monitors illustrating the current statistics. While OverSim has some visualization implementation, it is not as easy or as straightforward as Netlogo. This is mainly due to the fact that OverSim is result-oriented with little attention paid to the real-time visualization of the different interaction of nodes in the systems. However, NetLogo was built with the visualization aspect in mind. NetLogo programming language “logo” also has a very strong library of visualization commands, which allows the developer to demonstrate and visualize various interactions and mobility events effectively.

2.13. Summary

This chapter has provided an introduction to P2P networks, common P2P protocols, and the issues and challenges that exist in P2P networks and given a brief overview of network-modelling tools. A comparative analysis of conventional and ABM network modelling tools was provided for which GIA protocol from P2P networks was selected as the case study for implementation. The results were analysed for different metrics. Certain benefits of the results from ABM over those derived from conventional tools were observed while the limitations of both types of tools to model the complex networks were also noted.

The next chapter will provide a review of P2P worms and discuss the need for the development of a more realistic analytical model for worm propagation process. Also, a novel SEIR model will be proposed, which integrates important factors that affect the whole worm propagation process.
CHAPTER 3 : SEIR: A NOVEL ACTIVE WORM PROPAGATION MODEL

3.1. Introduction

This chapter provides an insight into existing P2P worms, their propagation behaviour and the type of vulnerabilities that they exploit from host operating systems/platforms. A brief description of the reported P2P worms will be provided, followed by a detailed review of the factors that affect the propagation process. Different analytical models, provided in literature, will be used with markedly different factors and different configuration settings. A brief classification of the existing models, based on parameters, will be presented. Taking into account the shortcomings of the existing worm propagation models, a stochastic discrete-time active worm propagation model will be proposed. A detailed description of each of the factors addressed in the proposed model will be given.

3.2. Worm Propagation in P2P Networks

A P2P worm is a software program that has a malicious code which takes the network as its medium of spreading. There are many reasons that make P2P networks vulnerable to the worms. One major reason is that many P2P networks, such as Kaaza, Fasttrack and eDonkey, are able to share executable files [78]. The second reason is that at any given time a huge number of common users are connected to P2P networks. These users are normally not aware of the security concerns related to P2P networks.

Worms and viruses are often treated interchangeably. However, there are technical differences between them [79]. A worm self-propagates in the network without any external assistance or action. It exploits the loop-holes in operating systems or programming errors. While a computer virus is a code script, it attaches itself to a computer program. Such a virus is dependent on the actions of its host program. It does not propagate itself but rather it requires a human action to get launched. By design, a worm could be said to be a sub-class of a virus. However, the worm is not just parasite but is an entirely autonomous body in terms of its propagation.
Worm can be categorized into two major categories based on scanning behaviour. These categories are scan-based worms and topology-based worms. The scan-based worms scan the entire address space using different scanning techniques, such as random scanning or hit-list scanning [13]. Examples of these worms are Slammer/Sapphire, Code Red I v2 (Moore, Shannon, & Brown, 2002, Code Red II[80], Conficker[81], [82], Blaster [83], Witty, Sasser. These worms are not dependent on network topology or neighbourhood information and an infected host is able to infect an unlimited number of computers. However, topology-aware worms have a knowledge of the network they intend to propagate or, at least, have the capability to discover topological information about the network if they do not have it in advance. Typical examples of topology-aware worms are email worms and social networking worms, which attack an exposed vulnerability in the systems and get list of neighbours from victims and further attack other peers. Slapper worms are a classic example of topology-aware worms. An overview of different worms in provided below:

3.2.1. Code Red & Code Red II

Code Red exploits the buffer overflow vulnerability, reported in Microsoft’s IIS web server. The first occurrence of Code Red was reported in July, 2001. The worm checks the date of the infected system to be from the first to the nineteenth of the month. Random IP addresses are generated using a random number generator using static seed values. It is programmed to launch a Denial of Service (DoS) attack from the twentieth to the twentieth-eighth of the month.

Code Red II is an extended version of Code Red I which uses a dynamic seed value instead of static seed value for a random number generator for an IP list [84]. Code Red II has a greater impact than Code Red I due to the large volume of infected hosts and requests sent to different hosts. It also infects devices such as routers, switches, printers and DSL modems.

3.2.2. Nimda

Nimda [14] is a self-propagative worm that was initially thought to be Code Red I due to its web-scanning activity. E-mails containing attachments and web-based probes differentiate Code Red I from the new worm “W32/Nimda-A” which is commonly called a Nimda worm. This worm affects the Microsoft Windows operating system and the name was chosen because
it represents “Admin” spelled backwards. The worm can spread via the following three means:

- **E-mail**: One of the delivery mechanisms of the worm is e-mail containing the attachment “readme.exe”. The subject e-mail pretends to be an IP address spoofed from the infected host. As soon as the e-mail is previewed, the host becomes infected.

- **Web Server Attacks**: Code Red II worm creates a backdoor on the IIS web server. A Nimda worm can exploit the backdoor and attempt to gain control of the web server.

- **Open Network Shares**: Networks that share contents with users, do not apply security policies and do not restrict unauthorized access could be a target of Nimda. This then allows for the possibility of distribution within internal networks.

3.2.3. Slammer

Slammer (also called Sapphire) [85] spreads by exploiting the buffer-overflow vulnerability in the Microsoft SQL Server Desktop Engine (MSDE). It is in fact the fastest spreading worm in history and can infect more than 90% of vulnerable hosts within 10 minutes. The first outbreak of this worm was reported in January 2003. Although the vulnerability in MSDE was identified in July 2002 and its patch was provided, the outbreak nevertheless still caused 75,000 vulnerable hosts to be immediately infected and caused significant financial losses.

The distinctive feature of this worm is its propagation speed. A full scanning rate (of more than 55 million scans per second) is achieved within three minutes by the worm. The growth rate is slowed down because significant portions of the network have insufficient bandwidth to accommodate it.

3.2.4. Blaster

The Blaster worm first struck in August 2003. It exploits the bug in the Remote Procedure Calls (RPCs) library in the Windows operating system [86]. The worm creates backdoors by copying all the code from the Distributed Component Object Model (DCOM) library and then injects its own code. This malicious code further launches a denial of service (DoS) attack on the Windows update server by using Transmission Control Protocol (TCP) port 80 SYN
flooding. It also uses the backdoor mechanism from this example exploit to transfer the worm payload to newly infected systems.

3.2.5. Conficker

First appearance of Conficker worm is reported in November 2008 which immediately infected millions of system world-wide [78]. Basically, it exploits the vulnerabilities in Windows-based operating system and spread through different techniques such as domain generation algorithm, defence mechanism and local propagation. Consequently, it has infected millions of victims in the world and even now the number is still increasing. A detailed review of the Conficker worm will be provided in Chapter 5.

3.3. Modelling P2P Worms

P2P networks have attracted significant attention due to their sharing of valuable data. The equal-sharing theme not only distributes the contents among different peers but also helps to handle any failover situation, which is the issue of conventional single-server-based content-sharing. While P2P networks provide unique benefits, they still pose a serious threat to social and economic domains. There are certain emerging issues and challenges that need to be addressed through focused research efforts. One of these issues is modelling the propagation of worms in a large P2P network, which consists of wired and wireless devices. Another issue is investigating the behaviour of worm propagation in P2P network considering different protocols such as GIA and Gnutella. Continuing the investigation of these issues, factors such as the effects of user behaviour, node mobility, configuration diversity and infection time-lag on the propagation of worms will be analysed within the context of P2P networks.

3.3.1. Worm Propagation Models and Classification

Worms in P2P networks pose a serious threat to all peers in the network. A worm attack can make a large portion of the internet unavailable. To take appropriate safety measures, apply efficient patching of nodes and ensure proper immunization of the influential nodes, it is essential to fully understand the propagation behaviour of worms [13].

P2P worms can generally be categorized into three categories based on their propagation method. Passive P2P worms usually copy themselves in the shared folder and then spread by
luring users to download and execute them. Reactive worms spread by means of exploiting vulnerabilities that exist in the software. They are spread only in the case when other hosts request files from the infected hosts. Proactive worms also spread through vulnerabilities, but they attempt to infect other hosts as quickly as possible with the help of neighbourhood information. An example of proactive worms is Code Red [87], Code Red II [80], Nimda, and more recently Slammer [88] and Blaster [83] and the Conficker worm [81], [82]. The Hybrid worm is a new hypothetical worm that could emerge in the future. The worm can adopt two or more propagation behaviours based on the circumstances. The existing research revolves mostly around these types of worm. However, proactive and hybrid worms have recently been gaining more attention from researchers.

The approaches followed in this research are based on developing analytical models and generating simulations of the worm propagation process in order to understand the behaviour of the worm. This helps when applying security updates, immunization strategies and recovery options. However, an analysis of large-scale networks with millions of nodes is practically impossible [5]. The analytical and simulation models are inspired by epidemiological disease spread models such as SI (Suspected-Infectious), SIS (Suspected-Infectious-Suspected) and SIR (Suspected-Infectious-recovered).

![Simple SIR Epidemic Model](image)

**Figure 3.1: Simple SIR Epidemic Model [89]**

The need for the development of analytical models has arisen due to one or more of the following reasons [79]:

1. To get a deep understanding of worm propagation behaviour;
2. To analyse and predict loss in cases of worm outbreak;
3. To evaluate network parameters such as node mobility, user behaviour, network link quality, configuration diversity and the degree of peers that affect the worm propagation;
4. To characterize how the worm infects the node and the device itself;
5. To devise immunization/patching/countermeasure strategies to contain and mitigate the outbreak.

A review of the existing worm propagation models in P2P domain will now be provided. These models have been classified according to the type of propagation adopted by the worm.

3.3.1.1. Active Worm Propagation Models

Different models exist in the literature and are based on the reasons for the development of analytical models mentioned in the previous section. A stochastic, discrete time model for active-worm propagation is based on the SIR epidemiological model, proposed in [7], and called STAWP model. This model does not consider the time a node takes to get infected but the unit infection time is nevertheless considered during the simulations. The major contribution of this work is to consider the dynamicity of network topology during active-worm propagation. The model is stochastic in considering a random value representing the number of nodes joining and leaving the networks. This random value is selected from the Probability Density Function (PDF) generated using Gaussian distribution. It assumes the infection time lag as the unit time, which is an important parameter in the infection process, as described by [67]. The impact of other important factors, such as node mobility, configuration diversity and user-behaviour, however, they are not considered in this model. The results presented in [90] are not more accurate than those from the Extended Topology Logic Matrix (ETLM) model due to the less supported precision of the floating-point values, which are computed during the validation of the analytical model. Since the floating-point part consists of more than 40 decimal places, a rounding-off of decimal places to achieve results with the available hardware resources is not possible. On the other hand, ELTM models are based on matrices and the large number of rows and columns of matrices make computation quite complex and practically impossible within the available computational resources [7]. A logic matrices-based approach is used to develop an analytical model for the propagation of worms from the difference equations of the logic matrix [91].

A discrete-time worm propagation model is proposed in [92]. The model (SEIRS) is an extension of the SIR epidemiological model with additional exposed state and cyclic behaviour (Figure 3.2). The model is assuming that a peer, once it is recovered, can revert to the suspected state again. However, this is a strong assumption because the peer goes to
recovered state only after particular operating system patches updates are applied or antivirus/antimalware signatures are updated for particular worm. This model also addresses the heterogeneous behaviour of nodes. However, this model provides partial diversity by dividing the network into different sets of monoculture nodes. Moreover, no attention is given to node mobility and user-behaviour. The infection time-lag factor is addressed in [67] based on previous work by [93]. The model is constructed based on Four-Factor model. These factors are (1) the human countermeasures, (2) the configuration diversity, (3) P2P topology and (4) the attack and defence strategies and are based on assumptions of a static network topology and a constant configuration diversity. Thus, this model does not address the infection time-lag and mobility of nodes.

Figure 3.2: SEIRS Model [94]

The author in [95] discuss the impact of node joining and leaving behaviour on the propagation ratio of topology-aware active worms. It shows that network size is independent of infection ratio and that the out-degree of nodes plays a vital role in worm propagation as the infection ratio increases to 90% as the node degree increases from 2 to 3. It also highlights the fact that the number of initial infectious hosts does not have a significant effect on the steady state propagation ratio.
Moreover, a ternary-matrix based approach is used to model the behaviour of active worm [96]. This work represents the overlay topology in the form of matrices, knowing the state of each node at any instant of time and the backtracking of infection from infected node to the infection-outbreak node. However, the matrix operation for large topologies containing hundreds of nodes does become difficult.

A discrete time deterministic propagation model of topology-aware worms is proposed in [97]. The worm propagation process is performed as a sequence of matrix operations by considering random graph topology and power-law topology. An extension of this approach for structured P2P networks considering different factors is proposed in [91]. However, the matrices used as the foundation of the analytical model are inappropriate for a large set of parameters and complex mathematical operations.

Agent-based modelling (ABM) is used as a tool to reflect the active worm propagation behaviour in [98]. This study considers the configuration diversity in relation to the degree of network nodes to reflect the immunization offered by such a combination. The complex behaviour of the worm in the presence of these factors is represented by taking advantage of the ABM tool. However, the ABM simulations require following the standards of modelling, such as, for example, ODD [99]. To achieve better results, it is suggested to follow the standards of modelling proposed by the modelling tool e.g. Netlogo proposes ODD as standard. The accuracy of the results also depends on the average values of multiple executions, which can be achieved in Netlogo using the Behaviour Space tool [100].

A hypothetical worm for BitTorrent (BT) networks is proposed in [101]. The impact of different parameters such as the Time-to-Live (TTL), the number of peers, the bread-first search and the depth-first search is evaluated. The behaviour of the worm is self-stopping after destroying a large fraction of the nodes in the network.

An SEIR model is developed in [102] with the focus on finding network conditions to achieve a malware-free equilibrium. An active worm affecting the file-sharing network is considered as an example for the study. The results indicate that users’ online/offline behaviour and the degree of nodes are important factors that can help to achieve a malware-free equilibrium.

Furthermore, the impact of network size on worm propagation is studied in [103]. It considers a two-layer epidemic model: the first layer considers all the networks and sub-networks in the
wild while the second layer deals with individual hosts. Its results indicate that the worm propagation process is slow at the initial stage but speeds up after some time and follows a short exponential tail at the end.

### 3.3.1.2. Passive Worm Propagation Models

Passive-worm propagation in file-sharing P2P networks is discussed in [104]. Considering the network as static and eMule [105] as protocol, they conclude that passive worm propagation can be reduced by decreasing the download rate before the corresponding patch is applied. An analytical model of the propagation of passive worms addresses binding, configuration diversity, online/off-line behaviour and the download duration of the file [106]. The results indicate that the worm files increase rapidly but, after a defined value, remain stable. The configuration diversity of the network is inversely proportional to the worm propagation and the increased values of diverse nodes reduce the infection density peak value.

The performance of passive worms over unstructured P2P networks is evaluated in [107]. The analytical model is developed based on the online/offline rates of the user and death rate of the nodes. The impact of different time delays during the different states of nodes is observed and has been shown to be the critical parameter in the epidemic spreading process.

Network throughput is considered as a factor impacting on passive worm propagation in [108]. This model was developed based on the SEI epidemiological model and further suggests the worm-defence strategy to be in accordance with file popularity, following Zipf distribution.

These limited studies indicate that user online/offline behaviour is a key factor in modelling passive worm propagation models. Passive worms have received relatively little attention, compared with proactive or reactive worms.

### 3.3.1.3. Hybrid Worm Propagation Models

A hybrid worm propagation model based on the Bluetooth channel and SMS/MMS services is proposed in [109]. The propagation model extends the SIR model to SEIRD (susceptible, exposed, infectious, diagnosed, and recovered). The foundation of this model is based on two-dimensional cellular automata. However, the factors affecting the propagation of hybrid worms and the impact of node mobility are not considered in this work.
3.3.1.4. Mobile Models

SMS worm propagation based on the social network graph is modelled in [110]. The RTSS (Reputation, Trust, and Susceptible) model is topology-aware and discrete-time model. This model considers the node reputation (R) and edge trust (T) degrees while SS refers to two susceptible (S) states.

An analytical model for worm spread via SMS and Blue-tooth channels is proposed in [111]. This model is an SI epidemiological model and demonstrates the simplistic behaviour of mobile malware spread. However, wireless-environment-related parameters such as channel contention, interference and path fading are ignored when modelling the worm propagation process.

The Agent-Based Malware Modelling (AMM framework) [112] is composed of a collection of networked and autonomous decision-making entities called agents. Any network device such as a mobile phone, a laptop, a server, an access point or a PDA is represented as an agent. The AMM framework is generic and represents many aspects of communication. However, we shall only consider P2P networks. AMM assumes that an agent could take place in some logical topologies corresponding to various services like e-mail, online messaging, P2P and social network. The agents are collected together in hierarchical manner. Few agents behave as access points in the wireless LAN and collect information from other devices, which reflect the real-life processing of information in such environments. The frameworks employ both mobility models, namely, Gauss-Markov (GM) and Random Way Point (RWP).

A study about the maximum possibility of infection in Mobile-to-Mobile communication is detailed in [113]. By utilizing the Pontryagin’s Maximum Principle (PMP) for the optimal control theory, the damage caused by a worm in a given network can be quantified. This study shows the impact of the mobile-device battery on infection and demonstrates that the infection is effective at the initial phase of an outbreak but and reduces as a function of time.

The worm spread of worm in a mobile ad-hoc network is considered in [114]. This framework considers SIS model as an epidemiological model and claims that any mobility model can be integrated with the framework using a set of adjacency matrices. Node velocity is also found to be an irrelevant factor in worm propagation in wireless.
The impact of the time-zone of the location of the peer in worm propagation and of the modelling P2P worm in social networks and in mobile networks are suggested as future directions for research in [13]. In the light of the recent increase in smart-phone users,[79] highlight the need for a general worm propagation model instead of a protocol- or worm-specific model.

Different types of propagation models have been reviewed. It has been noted that most of the work reported in the literature is oriented towards the modelling propagation behaviour of active worms. Both fixed and mobile networks are targeted by the research community. A list of the factors addressed in different analytical models can be seen in Figure 3.2. Contrary to what one would expect, very less attention has actually been paid to passive and hybrid worms and their propagation behaviour. A possible reason for this is that the quick spreading behaviour of active worms can cause a huge damage before any preventive measure can be taken. Moreover, passive worms are usually dependent on human actions and, as a result, trick users into downloading vulnerabilities from the internet and executing them [106]. Therefore, passive worms spread more slowly than active ones. The presence of hybrid worms is not

---

**Figure 3.3: Classification of Worm Propagation Models and Addressed Parameters**

Different types of propagation models have been reviewed. It has been noted that most of the work reported in the literature is oriented towards the modelling propagation behaviour of active worms. Both fixed and mobile networks are targeted by the research community. A list of the factors addressed in different analytical models can be seen in Figure 3.2. Contrary to what one would expect, very less attention has actually been paid to passive and hybrid worms and their propagation behaviour. A possible reason for this is that the quick spreading behaviour of active worms can cause a huge damage before any preventive measure can be taken. Moreover, passive worms are usually dependent on human actions and, as a result, trick users into downloading vulnerabilities from the internet and executing them [106]. Therefore, passive worms spread more slowly than active ones. The presence of hybrid worms is not
reported in the literature. However, analytical models have already been developed that portray different propagation behaviours in fixed and mobile networks [109].

The impact of passive, reactive and proactive worms being non-scanning worms is discussed in [115]. The results are generated based on a simulation for three different types of worm. These results indicate that the proactive worm is the most dangerous in terms of spreading activity while the impact of the passive worm is minimal, compared to that from other worms.

3.3.2. Immunization Strategies for P2P Worms

Immunization is a strategy to prevent a node and all its neighbours from being attacked by malicious software [116]. In the case of a P2P network, an efficient immunization strategy becomes important where worms can propagate quickly due to high connectivity and to the worm behaviour itself. Various immunization strategies have been developed and described in the literature dealing with P2P worms.

An immunization strategy for a social network worm is proposed in [117]. Although finding the most influential nodes in complex network is an open issue, attempts are still being made to find the key nodes based on vertex degrees, between-ness centrality, closeness centrality, Katz Centrality and the minimum dominating set. It is argued that the selection of these parameters for influential node selection rather choosing random influential nodes improves the patching scheme results. It is further suggested human behaviour when using a social network be incorporated in analytical models in order to estimate the worm-propagation behaviour more accurately.

The impact of heterogeneous immunization on worm spread is evaluated in [118]. Based on the SIR epidemiological model, two different types of immunization levels are assumed in this work. The results achieved indicate that most of the nodes maintain strong-immunization in order to reduce worm propagation.

The immunization strategy discussed in [119] is based on choosing the top ‘k’ influential nodes by voting. The election is done based on factors such as a social relationship graph and the node degree. However, the authors explain that more realistic parameters of social behaviour are required to more fully understand worm propagation and thereby the containment. Models of both Bluetooth and SMS/MMS worms are proposed using two-dimensional cellular automata in smartphones [120]. The behavioural characteristics of peers
are studied in [102], based on the SIR model. It shows that the bound on the spectral radius of the network needs to be revised. Furthermore, the user behaviour characteristics (e.g. offline-to-online transitional behaviour) and the communication patterns (e.g. neighbourhood size) of node result in an over-estimation of the epidemic threshold metric.

### 3.3.3. Stochastic versus Deterministic Behaviour of Worm Propagation

The stochastic and deterministic models can be divided according to the type of outcome they predict. The same output from given inputs is expected from a deterministic model which does not handle the random variation. On the other hand, the model may be more statistical in nature and so may predict the distribution of possible outcomes. Such models are said to be stochastic [121]. The deterministic models are not stable for even small variations in inputs and thus predict only well-defined cycles. The stochastic model predicts the extinction of at least one type for large populations.

If analysis is possible with many variables, there are then several reasons to develop a stochastic model [121]. Firstly, the natural way of spreading worms is stochastic, and there is always a chance of worm transmission from one individual to another. Secondly, the deterministic models are based on the law of large numbers but there are nevertheless phenomena that are stochastic by nature. For example, in a large community, a minor outbreak can infect a small portion of the network nodes while a major outbreak can indeed infect more or less than that. This does not reflect deterministic behaviour and a representation of the process is only possible through stochastic settings. These settings are a function of parameters that affect the worm propagation process. These parameters, such as user-behaviour, node mobility, and node degree, are computed randomly through probability distribution functions (PDFs). The third important point is about estimation. A stochastic model is required when there is some knowledge of uncertainty in the estimates. An estimate is invalid if there is no knowledge about uncertainty.

### 3.4. Factors Affecting the Worm Propagation Process

Different factors have been considered in the literature about the various analytical models that explore the different aspects of the worm propagation process. To overview on the
utilization of important factors in the proposed stochastic model, a brief and explicit review of each factor is now provided.

- **Configuration Diversity:** The heterogeneity of nodes is a network in terms of operating systems, software applications and configuration settings and can be referred to as configuration diversity. It becomes important when considering the impact of propagation dynamics at an individual user level or at a node level. Configuration diversity values are assumed to be constant in the existing analytical models rather than focusing on the mechanism of generating them from a given network [67] [93]. Configuration diversity also refers to the application installed on different platforms that may be running similar services. These nodes are not equally vulnerable because the same software may have different versions [122].

  Different researchers have modelled the phenomenon in different ways. Configuration diversity is modelled as a message-level interaction among users, or as a package installed on the host operating system [122]. Another study has indicated that software monoculture among nodes in a network can raise the security issues [92]. Diversification of client applications, the operating environment and software packages reduce the risk of worm propagation [92]. A colouring algorithm is used to represent the different sets of packages installed on different nodes in order to compute the configuration diversity. A unique colour is assigned to each node based on the number of packages installed on the node [92]. However, this theoretical foundation is not based on any statistical values generated by a real network. For example, the configuration diversity can be modelled more realistically pertaining to the fact that 75% of the nodes in P2P network are homogenous and they are configured with the same P2P client and operating system [123].

- **User Behaviour:** This is a broad term which includes the online/offline behaviour of nodes, the setting up upload/download limit by P2P client users, the free-rider problem and many more user behaviour issues. During the worm propagation process, different P2P clients log on and log off frequently on the nodes. Similarly, the upload and download limit of P2P clients is changed frequently.

  Different studies identified user-behaviour as random value [124][125]. However, Weaver [125] quantified user behavioural issues using statistical methods. These
studies indicate that, on average, 30 connection attempts by a single machine are made in a network infected by Conficker [125].

Preliminary studies show that the time duration needed for the propagation of passive P2P worms to reach a steady state is hundreds or even thousands of hours [6]. This work demonstrated files downloading and execution behaviour for passive worms. However, the analytical model partially addressed user-behaviour phenomena by not providing the logical statistical foundation.

The behaviour of users for checking short messages in social networks was modelled by introducing two susceptible states [110]. Empirical models are similarly used to model user behaviour in [112] and the results are based on the SMS data collected over a three-week period from a large scale cellular carrier with over 10 million mobile users. User behaviour is modelled as the offline/online behaviour of users in [126] where the probability of being online is simply considered to be a ratio of online users to the total number of users.

It is observed that the user behaviour is modelled only partially while its important components such as the upload/download limit and the free rider problem are ignored. Moreover, the statistical foundation and theoretical background could not receive the appropriate attention in order to achieve more realistic results.

3.5. **Requirements for a New Stochastic Analytical Model of P2P Worm Propagation**

Configuration diversity in P2P networks is addressed as a parameter in analytical models provided in [92][67][93]. Performance improvement mechanisms such as dividing the traffic load according to the capacity are proposed in [127]. The use of trusted-networks is similarly proposed by [128] for performance improvement. The impact of the behavioural characteristics of users and communication patterns in P2P networks are studied in [102][112][110]. The infection time-lag can be defined as the time a node takes to get completely infected from the starting-time of infection. The models presented in [92][67][129] considered the infection time delay involved during the worm propagation process.

The analytical models developed for modelling the worm propagation process are either stochastic or deterministic. The stochastic behaviour of worm propagation is reported by the
analytical models provided in [7][89][130][121]. However, the deterministic models have received relatively more attention in the literature than the stochastic models. The simple analysis of factors is the main advantage of deterministic models. A stochastic model needs to be developed with simple and relatively fewer parameters in order to be more mathematically manageable, but the results would still not be entirely realistic [121]. Deterministic models are solvable, even if they are more complex than stochastic models, where numerical solutions suffice [121].

Following these arguments, the need for the development of a generic worm propagation model is evident. This model should address and include important factors such as the infection time-lag, the configuration diversity and the user behaviour. The randomness characteristic of user-behaviour parameter makes the model stochastic.

As a result of the above-mentioned arguments, discussions and findings, the major research question addressed in this chapter can be stated as follows:

- What is the need/potential utility of introducing a discrete-time, stochastic, epidemiological model for the propagation of active worms that takes the infection time-lag, the user-behaviour and the configuration diversity of nodes into consideration?

3.6. Proposed Discrete-Time Stochastic Active Worm Propagation Model (SEIR Model)

There are important points to be considered prior to the development of the analytical model.

1- The model should consider active worm propagation in unstructured P2P networks.

2- The model should assume worm propagation to be a stochastic process that depends on factors such as infection time-lag, user behaviour and configuration-diversity.

3- The extended SIR model that is SEIR should be considered during the development of the analytical model in order to address the infection time-lag phenomena. Each infectious node has a specific capacity for infection which is directly proportional to the infection time-lag where the infection time-lag can be defined as the time that a node takes to get infected as soon as it gets exposed [90].
The major focus of this model is to study holistically the factors, namely, the configuration diversity of nodes in a network, the impact of variation in user behaviour and of the infection time-lag on worm propagation. All of these factors are assumed to be independent variables that affect the process of worm propagation. A description of different states is provided as follows:

- **Susceptible**: these nodes are part of network and susceptible to infection.
- **Exposed**: it is a state of nodes when they get exposed of worm. The node is not infect yet and worm may take time from few seconds to number of hours in order to infect the machine. This time depends on many factors such as node configuration or the worm behaviour itself.
- **Infected**: the state of a node in which it has been infected from the worm and has done the damage.
- **Recovered**: it is the last absorbing state. From any of the other states, a node can go to recovered state depending on that stage specific settings.

A graphical representation of this is shown below in Figure 3.4:

![Figure 3.4: The States and Transitions](image)

The recovered state is the absorbing state and is represented by “R”. Initially, in steady state, all the nodes are in the susceptible state. There are four states and five state-transitions. These transitions are explained below:

1. A node can transit from a suspected “S” to a recovered “R” state. The rate “A” depends on the configuration diversity, the user-behaviour and the degree of node where the degree refers to number of neighbours of the nodes.
2. A node can transit from a suspected “S” to an exposed “E” state at rate “B”. The rate is dependent on the configuration diversity, the user-behaviour and number of neighbours.

3. The exposed to the infected state transition is only dependent on the birth and death rate of nodes in the network.

4. The infectious-to-recover transition is done at rate “γ” and the number of nodes transiting from the infectious-to-recovered state at time t are equal to γ I (t).

5. The node can transit from exposed to a recovered state at the rate “E”. The factors that can affect the state transition are configuration diversity and node mobility. However, a discussion on node mobility is beyond the scope of this chapter. It is important to note that once a node get exposed, it will eventually be infectious even if disconnected from the network due to mobility. However, the disconnected infectious node does not have an impact on worm propagation in the network.

The state transition also takes fractions of a second to execute where a human counter measure is impossible within this minimal period of time. The time of the state transition varies from node to node due to the variation in operating systems, software packages and antivirus programs.

3.6.1. Notations

The symbols and notations used to describe the proposed model are provided in Table 3.1:
Table 3.1: Notations and Description

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>Total number of nodes in the network</td>
</tr>
<tr>
<td>( S(t) )</td>
<td>Number of suspected nodes at time ((t))</td>
</tr>
<tr>
<td>( E(t) )</td>
<td>Number of exposed nodes at time ((t))</td>
</tr>
<tr>
<td>( I(t) )</td>
<td>Number of infectious nodes at time ((t))</td>
</tr>
<tr>
<td>( R(t) )</td>
<td>Number of recovered nodes at time ((t))</td>
</tr>
<tr>
<td>( \delta )</td>
<td>Configuration diversity. (0 \leq \delta \leq 1)</td>
</tr>
<tr>
<td>( \beta )</td>
<td>User behaviour. (0 \leq \beta \leq 1)</td>
</tr>
<tr>
<td>( K_i )</td>
<td>Degree of node (i)</td>
</tr>
<tr>
<td>( \varepsilon )</td>
<td>Infection time lag</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>Infectious to removal rate</td>
</tr>
<tr>
<td>( \tau )</td>
<td>Rate of transition from exposed to infectious state</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Rate of transition from exposed to recovered state</td>
</tr>
<tr>
<td>( b )</td>
<td>Birth rate of nodes in the network</td>
</tr>
<tr>
<td>( d )</td>
<td>Death rate of nodes in the network</td>
</tr>
</tbody>
</table>

3.6.2. Derivation of Factors and Sub-Factors

3.6.2.1. User Behaviour

The user behaviour includes the setting-up of P2P clients, of the upload limit, the download limit, the user offline/online behaviour and the custom setting for data sharing. User behaviour varies from client to client and from application to application [131]–[133]. These are user-defined parameters that affect P2P client and hence the worm propagation process. User-behaviour may include user actions that a user performs with P2P client. These actions are dependent on the type of client application. If the client application is for video streaming, the client may select priority videos or search for specific keywords. If the client application is for a file-sharing client, the user may want to set an upload and download limit for the file. The on/off behaviour itself of users for applications or nodes also plays a major role.

Considering that user-behaviour is only dependent on the probability of user offline/online (\(\kappa\)), on the probability of setting up an uploading/downloading limit (\(\mu\)), on the probability of
human counter measures (π) and on free-rider probability (σ), it is important to note that μ, π and σ are mutually independent but dependent on online/offline probability, κ. As a result, the user-behaviour (β) can be computed as follows:

\[
\beta = P(\mu|k) \times P(\pi|k) \times P(\sigma|k) \tag{3.1}
\]

Since the upload/download limit set up by the individual user, the human counter-measures taken by an individual and the probability of an individual being a free-rider at any instant in time are considered as random variables their values are drawn from the Probability Density Function (PDF), generated using Gaussian distribution with given mean and standard deviation values. It establishes the fact that the user-behaviour (β) is a random variable for a given P2P node.

### 3.6.2.2. **Configuration Diversity**

With a given network topology, the configuration diversity can be computed using the Bucket theorem. This theorem is based on set theory and assumes number of sets initially equivalent to the number of nodes in the networks. Basically, each node belongs to a set and added in the sets one by one based on defined criteria. As soon as, all the nodes are added to the buckets (sets), the remaining buckets are discarded. This theorem assumes that the number of nodes in a network ‘N’ are known and each node has one operating system installed. This theorem disregards the software packages and utilities on the node. Initially, ‘X’ buckets are created and initialized with zero nodes in them where \( X \leq N \). Consider the following sample topology with \( N=10 \).
Each node is traversed and it can be seen that only 3 buckets have nodes while other buckets have zero items according to the number of different operating systems. By considering only the number of Windows-based clients (W), the configuration diversity can be computed as:

$$\delta = \frac{N-W}{N}$$  \hspace{1cm} (3.2)

The computation of the configuration diversity can be extended by considering the packages, updates and software development kits (SDKs) installed on the machine.

3.6.3. State Transitions and Factors

3.6.3.1. Suspected-to-Exposed Transition

The impact of parameters at time ‘t’ is given as:

- **User-behaviour (β)**: If \((1-\beta)\) is the state transition probability from the suspected-to-exposed state, then the probability to transit from the suspected-to-recover state is \(\beta\).

- **Configuration diversity (δ)**: It plays a major role in worm propagation. \((1 - \delta)\) is the probability of a node to go from the suspected-to-exposed state while \(\delta\) is the probability to go from the suspected-to-recovered state.
• **Infection time lag**: This factor is irrelevant from the suspected-to-exposed state and the suspected-to-recovered state [90][92].

3.6.3.2. **Exposed-to-Infectious Transition**

A node in the exposed state can transit from the Exposed-to-infectious state. The probabilities to transit from the exposed to the infectious state are discussed below:

- **User-behaviour**: Since only the propagation of active worms is considered, the user-behaviour becomes irrelevant in worm propagation for the exposed state. The reason is that active worms can take a fraction of a second to make the node infectious.

- **Infection time-lag**: Due to the different configurations of each node, the infection time-lag is different as well. The maximum time for a node can take to transit from exposed to infectious is $T$. There are $n$ intervals each with $t$ duration where $n \times t \leq T$ and $T \mod t = 0$. The infection time-lag is a function of configuration diversity and $\varepsilon \sim \delta$. Where $\varepsilon$ is the infection time-lag.

3.6.3.3. **Exposed-to-Recovered Transition**

A node can transit from the exposed to the recovered state. The factors that may affect the state transition are antivirus or antimalware programs installed on the machine.

3.6.3.4. **Infectious-to-Recovered Transition**

If the removal rate from the infection state is $\gamma$, so the number of nodes in the recovered state at time $t$ from infectious state is $\gamma I(t)$.

3.6.4. Model Assumptions

The assumptions that are taken to develop the analytical model are as follows:

1- Initially, all the nodes are in the suspected state.
2- Initially, the infection starting nodes are chosen randomly.
3- The value of the configuration diversity ($\delta$) is computed using the Bucket theorem (Equation 3.2) and the value of the user behaviour ($\beta$) is computed using Equation 3.1.
4- The factors ($\delta, \beta$) are independent of each other.
5- At any given time, a node can be in any of the four states that are susceptible $S(t)$, Exposed $E(t)$, Infectious $I(t)$ and Recovered $R(t)$, where $S(t) + E(t) + I(t) + R(t) = N$. 

---

88 | Page
3.6.5. Relationship among Parameters

- **Suspected-to-Recovered State**: ‘A’ is the rate of transition and ~ is the symbol of proportionality.

\[ A \sim (\delta) \quad (3.3) \]
\[ A \sim (\beta) \quad (3.4) \]

The infection time lag is irrelevant for this state transition. The value \( \delta = 0 \) means homogeneous network of nodes while \( \delta = 1 \) means perfectly heterogeneous network.

K is the degree of a node and \( P(k) \) is the probability of the K neighbours to go in the corresponding state. The higher the degree of a node, the higher will be the probability of neighbours going into the corresponding state. It is also important to note that the probability of infection increases with an increase in the degree of nodes.

- **Suspected-to-Exposed state**: ‘B’ is the rate of transition from the suspected to the exposed state.

\[ B \sim (1 - \delta) \quad (3.5) \]
\[ B \sim (1 - \beta) \quad (3.6) \]

- **Exposed-to-Infectious State**: The exposed interval maximum upper bound is \( T \) i.e. the maximum time a node takes to go from the exposed to the infectious state. It is divided into \( n \) subintervals of length \( t \). Any node can go form the exposed to the infectious state in \( n \times t \leq T \) where \( T \% t = 0 \).

In the exposed state, the infection time-lag is a function of the configuration diversity, so the relation \( \Theta = \delta \times T \). For the population that can go from exposed to infectious state is:

\[ I(t+1) = \tau E(t) - \gamma I(t) - dI(t) \quad (3.7) \]

Where ‘\( \tau \)’ is the transition rate from the exposed to the infectious state, ‘d’ is the death rate at the infectious state and ‘\( \gamma \)’ is the recovery rate, since the only factor affecting this transition is configuration diversity.
- **Exposed-to-Recovered State**: The number of recovered nodes at time \( t \) from the exposed state are

\[
R(t+1) = \rho E(t) + \gamma I(t) - dR(t) \quad (3.8)
\]

### 3.6.6. Complete System of Equations

Consider a network of \( N \) peers when the probability of a peer being attacked by at least one attacker is \( 1/N \). If \( (\beta) \) is the user behaviour, then the probability of a peer being attacked by one attacker is \( (1-\beta)/N \) and the probability of not being attacked is \( (1 - ((1-\beta)/N)) \). If \( K_i \) is the degree of the node, then the probability of not being attacked by \( K_i \) neighbours is \( (1 - ((1-\beta)K_i/N)) \).

Most empirical networks share common properties in terms of their local features. For example, some networks have very skewed degree distribution so that most individuals have few neighbours but there are a few exceptional individuals that have high degree distribution. The latter type of nodes is often called ‘hub’ or ‘super-spreader’ [134], [135]. In the context of P2P networks, it is called a ‘Super-Peer’. The skewed distribution is often modelled using a power-law where it is assumed that for some \( \phi \geq 1 \), \( P(k) \sim k^{-\phi} \) for large degrees, where \( k \) is degree of a node [136]. The case where \( 2 < \phi < 3 \) gained significant attention and is called scale-free distributions [136]. To calculate the value of \( \phi \) using networks structure, let \( \{K_1, K_2, K_3, \ldots \ldots \ldots, K_N\} \) be the degree of each node in the network and Mean, \( M = \frac{\sum_{i=1}^{N} K_i}{N} \). The super-spreading parameters, \( \phi \), can be computed as:

\[
\phi = \sqrt{\sum_{i=1}^{N} [K_i - M]^2} + 1 \quad (3.9)
\]

Where \( \phi \geq 1 \), it is to be noted that the mass action law applies with the homogenous mixing assumption that each individual has the same chance of coming into contact with an infectious individual [137]. This hypothesis eliminates the need to know the precise contact network on which the disease spreads. However, true populations violate some mass-action assumptions in a manner affecting the epidemic dynamics [138]. The true population is based on a random mixing assumption.

If a suspected node is attacked, it has the probability \( (1 - \delta) \) to be exposed. So the number of exposed hosts from time \( t \) to \( t+1 \) is calculated as:
\[ E(t+1) = S(t) (I(t))^\phi (1 - \delta) (1 - \beta) - \tau E(t) - \rho E(t) - dE(t) \] (3.10)

So the number of peers in the recovered state is:
\[ R(t+1) = \gamma I(t) + S(t) (I(t))^\phi (\delta)(\beta) - dR(t) + \rho E(t) \] (3.11)

The number of nodes in the infectious state is:
\[ I(t+1) = \tau E(t) - \gamma I - dI(t) \] (3.12)

The number of nodes in the susceptible state are:
\[ S(t+1) = -S(t) (I(t))^\phi [((1 - \delta)(1 - \beta) + \delta \beta)] + bN - dS(t) \] (3.13)

Where \( S(t) + E(t) + I(t) + R(t) = N(t) \) is true if and only if the birth-rate of nodes in the network is equal to the death-rate in the network i.e. \( b = d \).

3.6.7. Boundary Conditions for SEIR Model

The validity of analytical model is evaluated using convergence or boundary conditions. The boundary conditions validate the operating range of the analytical models. Convergence conditions for SEIR model are applied in order to understand the behaviour of system on boundary conditions.

For state transitions, at time \( t=0 \), \( S(0) = N \) and \( E(0) = I(0) = R(0) = 0 \). Also, any time \( t \), \( S(t) = E(t) = I(t) = R(t) > 0 \).

**Case 1:** The boundary conditions for equation 3.10 are described as follows:
\[ E(t+1) = S(t) (I(t))^\phi (1 - \delta) (1 - \beta) - \tau E(t) - \rho E(t) - dE(t); \ \forall 0 \leq \delta \leq 1, 0 \leq \beta \leq 1, \phi \geq 1 \text{ and } \tau, \rho \text{ and } d \in \text{ the range of } [0,1] \]

**Example:** Consider lower bound of parameter values for the exposed state equations converges the equations as follows:
\[ E(t+1) = S(t)(I(t)) ; \ \forall \delta = 0, \beta = 0, \phi = 1, \tau = 0, \rho = 0, d = 0 \]

Above equation represent the uniform spreading of worm.

Similarly, consider upper bound of parameter values for the exposed state equations converges the equations as follows:
\[ E(t+1) = -3E(t); \ \forall \delta = 1, \beta = 1, \phi > 1, \tau = 1, \rho = 1, d = 1 \]
The above equation indicates that the infection process stops completely. No new nodes are
going exposed while existing exposed nodes can go to infectious state or recovered state or
die out from the network.

**Case2:** The whole system of equations without applying any boundary condition is provided
in equation (3.14). Consider the boundary conditions as follows:

\[\forall 0 \leq \delta \leq 1, 0 \leq \beta \leq 1, \varphi \geq 1, \text{ and } \gamma, \tau, \rho \text{ and } d \text{ in the range of } [0,1]\]

Consider lower bound of parameter values for SEIR model converges the equations as follows:

\[\forall \delta = 0, \beta = 0, \varphi = 1, \tau = 0, \gamma = 0, \rho = 0, b = 0, d = 0\]

Consider upped bound of parameter values for SEIR model converges the equations as follows:

\[\forall \delta = 1, \beta = 1, \varphi > 1, \tau > 0, \gamma > 0, \rho > 0, b > 0, d > 0\]

The upper bounds of parameter values indicate that all the nodes in susceptible state will go
to recovered state and do not transit to exposed state. All the nodes in the exposed state will
die or transit to infectious or recovered state. All the nodes in in infectious state with die or
transit to recovered state with time.

At any time \(t\), \(S(t) < N\), \(R(t) < N\), \(E(t) = 0\) and \(I(t) = 0\) where \(S(t) + R(t) = N\).

### 3.7. Conclusion

In this chapter, an extensive analysis and review of the state-of the art of P2P worms in P2P
networks has been provided. Different analytical models addressing a variety of parameters
were studied and classified. These studies revealed that the important factors in the worm
propagation process are either neglected or only partially addressed in the analytical models.
This inspired the development of a new stochastic and discrete time SEIR model for worm
propagation modelling, which could holistically integrate the configuration diversity, the
infection time-lag and the user-behaviour in an integrated analytical model.

In the following chapter, exhaustive results based on the variety of values of the parameters
addressed in SEIR model will be provided. The proposed model will be evaluated using
conventional and ABM tools, by considering two common unstructured P2P protocols, GIA
and Gnutella. This will not only help in the comparative evaluation of different tools in terms of modelling but will also ensure the accuracy of the results and features provided by this particular modelling paradigm.
CHAPTER 4 : SEIR MODEL: SIMULATION RESULTS AND DISCUSSION

4.1. Introduction

A discrete-time analytical model (SEIR model) was proposed in Chapter 3 in order to reflect P2P worm propagation process and its associated factors. The proposed model addresses user behaviour, configuration diversity of nodes and infection time lag.

In subsequent sections, the implementation of SEIR model using a conventional network modelling tool and an ABM tool will be provided. To the best of our knowledge, this thesis provides the first integrated implementation of common, complex P2P protocols, such as Gnutella and GIA, along with SEIR model. The impact of a variation of different parameters will be observed, analysed and discussed in order to investigate their role in the worm propagation process and this will subsequently help in the development of immunization strategies.

4.2. Performance Evaluation of SEIR Model Using Conventional Modelling Approach

In this section, Matlab is selected as the modelling tool for the evaluation of the proposed SEIR model. Matlab is specifically designed to solve scientific and engineering problems. To express computational mathematics, it uses a matrix-based language. A library of built-in algorithms and codes provides the essential domain-specific support. Being based on a desktop environment, it provides support for experimentation, exploration, and discovery. The tool has been rigorously tested for its capabilities for modelling the worm propagation process.

The proposed (SEIR) model is evaluated by comparative analysis with two selected benchmark models, STAWP model [7] and with the Time Four-Factor model [93].

Since unstructured P2P networks have the properties of scale-free networks [139], for the implementation of these analytical models, the Barabasi-Albert Scale Free Network (BA –
SFN) [140] is used to generate the network topologies. All subsequent networks will be generated using the same approach. The details of the benchmark models used are as follows:

1- STA WP Model [7]: This model is a discrete time, active worm propagation stochastic model. It is based on the SIR epidemiological model that takes into consideration the impact of dynamic network topology on the worm propagation process while ignoring the infection time-lag and node heterogeneity.

2- Four-Factor Model [93]: This discrete-time analytical model has been paid attention in the literature because it handles four factors simultaneously. These factors are network topology, countermeasures considered by users and Internet service providers (ISPs), configuration diversity of nodes in the P2P network, and attack and defence strategies. It introduced an exposed state to the standard SIR model and modified the SIR model to SEIR. It proved that the infection time-lag of each node in an unstructured P2P network is different from that of other nodes due to configuration diversity. However, the impact user-behaviour of nodes is not specifically addressed in this model.

In the following section, the process of worm propagation will be discussed considering our proposed SEIR model in comparison with the benchmark models. It is important to note that in all plots in Section 4.2, the unit of time along x-axis is second. The results are presented for all the models, showing the influence different values of configuration diversity, infection time-lag and user behaviour.

4.2.1. Infection Propagation over Time

To measure the number of infectious hosts over time, a Scale Free Network (SFN) of 100 nodes is generated. All the nodes are considered vulnerable before infection starts. The infectious node is randomly selected from all the nodes in the network. The implementation of all benchmark models is carried out in Matlab considering two common unstructured P2P Gnutella and GIA protocols.

The infectious population density over time during the worm propagation process is shown in Figure 4.1:
The average value of 20 executions from STAWP model for the propagation of active worms demonstrates a maximum infection density of 90%. The removal rate from the infectious to the removed state is considered to be 0.05%.

The results are comparative with that in [7], whose results show peak infection density to be 80%, with a 0.05% removal rate. Such a high infection density demonstrates that the model does not consider the important factors that may affect the worm propagation process. Considering these factors may result in a lower infection density when compared with the one derived from STAWP model (Figure 4.1).

Infection propagation over time using Four-Factor Model is shown in 4.2:
The plot shown in Figure 4.2 is averaged out from 20 executions with a configuration diversity value considered to be 0.2 for all executions. This means that 20% of the nodes are different from each other in terms of operating systems, installed packages and P2P clients. The results show that in the presence of an additional parameter (configuration diversity), the worm propagation process decreases significantly, compared to STAWP model. 45% of the nodes are shown to be infectious which are order of magnitude less than that infectious nodes by STAWP model.

The plotting of infection propagation over time using SEIR model along with STAWP and Four-Factor Models is shown in Figure 4.3:
It can be seen that, ignoring configuration diversity and user-behaviour, the infection density tends to peak with its highest value being 30% for a network of 100 nodes. At the same time, the infection density, according to Four-Factor Model is more than 40%, which is due to the additional exposed state in SEIR model. It can be seen that the presence of different factors affects the worm propagation process. The more the number of parameters and factors, the higher will be the impact on infection density.

In the following sections, the impact of different factors on the worm propagation process considering different P2P protocols will be examined. A comparative analysis of the benchmark models and the proposed model is given in detail.

4.3. First Case Study: Comparative Evaluation of the Analytical Models Considering Gnutella Protocol

The impact of different factors on the worm propagation process using SEIR, STAWP and Four-Factor models will be examined. In this section, all the models are implemented using
Matlab with Gnutella being the underlying P2P protocol. A network of 500 nodes will be generated. It is assumed that all the nodes are vulnerable to infection with equal probability.

4.3.1. The Impact of Configuration Diversity on Infection Propagation

Configuration diversity is addressed by Four-Factor and SEIR models. The results showing the different values of configuration diversity and the impact on worm propagation are shown in Figures 4.4(a) and 4.4(b).

STAWP model shows a rapid increase in the worm propagation process, where it quickly reaches between 400 and 450 nodes but then drops exponentially. Four-Factor and SEIR models, on the other hand, show an exponential increase in node infection, reaching a maximum value of between 100-200 for a configuration diversity of 0.3 and 0.5.

There are two major differences between Four-Factor and SEIR models (Figure 4.4(a)). The infectious nodes Four-Factor model are almost 10% more than those of SEIR model. The reason for this is that SEIR model involves an additional exposed state, from which some nodes directly transit to the recovery state instead of transiting to the infectious state. This difference in the number of infectious nodes depends on the rate of transition to the recovery state from the exposed state. The second observation is the time-delay in SEIR model results, compared to Four-Factor Model results. The underlying reason is the infection time-lag, which occurs in the exposed state due to the different operating systems, packages and programs configured on the different nodes in the network.

Figure 4.4: Configuration Diversity for all Models
4.3.2. The Impact of User Behaviour on Infection Propagation

The impact of the different values of user-behaviour on infection density is observed in Figure 4.5(a) and 4.5(b). Since Four-Factor and STAWP models do not consider user-behaviour, the observed infection density, based on these models, is much higher than that from SEIR model. The results are generated by considering a homogeneous network of nodes i.e. the configuration diversity is equal to zero. The intent is to highlight the impact of this factor on the worm propagation process, as compared with other analytical models. It can be seen that the infection density is 20% less than that from Four-Factor Model with a value of 0.3 for user behaviour while it is more than 30% less than that from Four-Factor Model with a value of 0.5 for user behaviour. The reason for this is that SEIR model takes into account for user-behaviour as factor while other models ignore it.

![Graphs showing infection density over time for different values of user behaviour](image)

(a) User Behaviour = 0.3  (b) User Behaviour = 0.5

*Figure 4.5: User-Behaviour for all Models*

4.3.3. The Impact of Random Values on Infection Propagation

Since SEIR model considers user behaviour and configuration diversity as factors that impact on the worm propagation process in P2P networks, a heterogeneous network in which nodes are affected by user-behaviour is considered. The infection population density over time is computed in such a network and shown in Figure 4.6 along with the findings from Four-Factor and STAWP models. The results demonstrate that for a heterogeneous network in which users are continuously affecting P2P clients, the infection density is very less which is around 20-
30% of the total susceptible population. A drop in infection ratio of up to 30-40% can also be observed for Four-Factor model because it considers the impact of configuration diversity on only the worm propagation process. However, there is no impact of any of the considered factors on the worm propagation process for STAWP model and it shows a high infection density of more than 90%.

![Infection with time (random values)](image)

**Figure 4.6: The Impact of Random Values of Parameters**

4.3.4. Random Values of User-behaviour for Homogeneous Network

For a homogeneous P2P network of nodes where all the nodes are assumed to have a similar number of package and configuration diversity the value is zero. The users are operating P2P clients randomly, the infection propagation reach a maximum peak value of between 20-25% for SEIR model and then decreases, as shown in Figure 4.7. It can be observed that Four-Factor and STAWP models remain at the highest of their infection density values of 50% and 90% respectively. This is due to the fact that only Four-Factor Model considers configuration diversity, which is assumed to be zero due to the homogeneous network of nodes. While STAWP model does not consider any of the factors.
4.4. Second Case Study: Comparative Evaluation of Analytical Models Considering GIA Protocol

A scale-free network of 500 nodes is generated using the BA-algorithm. The features of GIA [16] such as biased random walk, one-hop replication and active flow control are also implemented. The capacity of each node is assigned based on the number of neighbours the node has. The infection starting-point is selected randomly in the network. In the following section, the infection density over time is derived using different parameter values and then evaluated with selected benchmark models.

4.4.1. The Impact of Configuration Diversity on Infection Propagation

The results of different values of configuration diversity and their impact on worm propagation are considered in this section. A significantly faster worm propagation process is observed in P2P network based on GIA protocol as compared with the network based on Gnutella protocol. The number of infectious nodes is between 30-40% with configuration diversity values of 0.3 and 0.5 for SEIR while the number of infectious nodes is more than 70% for Four-Factor Model with similar values of configuration diversity. There is no impact of configuration diversity on STAWP model (Figure 4.8).

It can be noted the trends so far observed that Four-Factor and STAWP models are of an exponential nature while SEIR model shows relatively linear trends. Another observation is
that the infection time increased, compared with that for Gnutella under the same assumptions (Figure 4.4). The reason for this is that the additional parameters, such as user-behaviour and infection time-lag in SEIR model, make the worm propagation process relatively slower, thus representing relatively realistic view as compared with that provided by Four-Factor and STAWP models.

4.4.2. The Impact of User Behaviour on Infection Propagation

The impact of different values of user-behaviour is shown in Figures 4.9(a) and 4.9(b) for SEIR model in comparison with STAWP and Four-Factor models. The difference in infectious population following all models is shown in Figure 4.9. The infectious population following SEIR model is much less than the infectious population using Four-Factor and STAWP models. The reason is that both Four-Factor and STAWP models do not consider user behaviour as a parameter that can affect the propagation process.
4.4.3. The Impact of Random Values on Infection Propagation

In this section, all of the various factors (user-behaviour and configuration diversity) are randomly considered for simulating worm propagation. A relatively lower infection density is observed following SEIR model under the given values, compared with Four-Factor and STAWP models.

It can be observed in Figure 4.10 that, for SEIR model, the infection density reaches a maximum value of 30% -35% due to the handling of random values (in a range of 0 to 1) of configuration diversity and user behaviour. However, infection density is near to 60% for Four-Factor and 90% for STAWP model.

While other observations were discussed in Section 4.3.1 and Section 4.3.2, the drop in the curve starting from 75 to 90 units of time is approximately the same for all the analytical models. It shows that when most of the nodes in the network get infectious, the worm propagation trends are similar for all analytical models regarding the active worms.

![Figure 4.9: User-Behaviour for all Models](image-url)
4.4.4. Random User-Behaviour Values for Homogeneous Network

With a homogeneous network, considering the random values of user-behaviour (in the range of 0 to 1), the infection density over time for SEIR, STAWP and Four-Factor models is shown in Figure 4.11. It can be seen that the random values of parameters reduce the infection density significantly, compared to Four-Factor and STAWP models. However, the infection population over time for Four-Factor model in the absence of configuration diversity is higher (Figure 4.11).
4.5. SEIR: The Impact of Different Parameter Values

The proposed SEIR model is evaluated for all the parameters (configuration diversity, infection time-lag and user-behaviour). The impact of variation of the parameter values on infection propagation is discussed considering Gnutella protocol. A scale-free network for 500 nodes is generated and the infection starting-point is chosen randomly among all the nodes in the network.

4.5.1. The Impact of Configuration Diversity on Infection Propagation

Different values of configuration diversity are considered and infection propagation graphs are recorded for each value. The observations are summarized in Figure 4.12 below:

The values of configuration diversity are considered to be 0.3, 0.5 and 0.8 to evaluate the impact on infection propagation using SEIR model. A significant decrease in infection propagation is observed with the increasing value of configuration diversity in the network (Figure 4.12). It also highlights that the homogeneous network can more rapidly serve as a medium for infection propagation.

A node could be different from any other node in terms of the packages, updates or service packs installed on it. Anti-malware and antiviruses could also be configured on nodes as well. The programs on a node can work as immunizers, because worms are designed to exploit certain vulnerabilities in the operating system. If the vulnerability is patched or an antivirus is
configured, the node not only immunizes itself but also stops the worm propagation process at this point in the network. A node in a susceptible state can transit from a susceptible state to an exposed state or a recovered state. For a high value of configuration diversity, more nodes transit to recovered state instead of to an exposed state.

4.5.2. The Impact of User Behaviour on Infection Propagation

SEIR model considers user-behaviour an important parameter and contends that it impacts the worm propagation process in different states. The impact of different values of user-behaviour on infection density is therefore shown in Figure 4.13:

![Figure 4.13: Impact of User Behaviour over Time on Infection Propagation](image)

Similar to the configuration diversity variation results (Figure 4.12), user-behaviour can be seen to have a significant impact on the worm propagation process. The results confirm the claim of analytical models that higher user actions with P2P clients slow down the worm propagation process [141].

4.5.3. Random Parameter Values for Different Network Sizes

The random values of configuration diversity and user behaviour are considered in this section. The results are computed for homogeneous networks of 100, 300, 500 and 1000 nodes. The user-behaviour and configuration diversity values are computed randomly in the range of 0 to 1. The results of infectious population density over time for four different network sizes are given in Figure 4.14:
The results are computed with an average of 20 executions for all networks. The results show the infection density to be 25% for 100 nodes, more than 30% for 300 nodes, 36% for 500 nodes and 37% for 1000 nodes. These results indicate that infection density increases as the network size grows, irrespective of the fact that the nodes experience different configuration diversity and user behaviour values. The reason for this increase is the alternate query routing options available to the node.

4.5.4. The Impact of Super-Peers on Infection Propagation

Super-peer is defined as a node with which large number of nodes are connected [142]. Super-peers are not explicitly defined in Gnutella while the feature of topology adaptation based on capacity generate super-peers in GIA. This definition is valid for Gnutella and causes scalability problems as well. To counter the problem, GIA protocol define super-peer as a well-connected node along with high capacity as well.

The impact of super-peers on the worm propagation process can be seen in Figure 4.15. The infection population density over time for SEIR model is measured for Gnutella and GIA.
protocols without considering any value for configuration diversity and user-behaviour parameters. The infection population density is found to be 15% more in GIA as compared with that in Gnutella. The reason is the biased random walk towards high capacity and the connected peers in GIA, compared with a random walk with no super-peer in Gnutella.

![Impact of super-peer](image)

**Figure 4.15: The Impact of Super-Peer on the Worm Propagation Process**

### 4.6. Discussion

The impact of the configuration diversity of the nodes in P2P network has been observed for STAWP, Four-Factor and SEIR models, considering Gnutella (Figure 4.4) and GIA (Figure 4.8). It can be seen in Table 4.1 that the infection propagation for GIA protocol is higher than for Gnutella protocol under the same values of configuration diversity. It can also be seen that STAWP and Four-Factor models show relatively higher values of infection, compared to those shown by SEIR model under both protocols. The reason for this is its efficient query-forwarding mechanism due to the biased-random walk in GIA, compared to Gnutella. Another reason is the super-peers created in GIA due to a dynamic topology adaptation feature. These super-peers have the capacity to handle query requests, thus solving Gnutella scalability issue and achieving efficient query propagation.
Figures 4.4 and 4.8 show a delay in the curve of SEIR model, compared to Four-Factor model; this is due to an additional exposed state that SEIR model reveals. There is a quicker drop in the infection density when using SEIR model compared with Four-Factor model because some of the nodes go to a recovered state in SEIR model due to the high value of configuration diversity.

### Table 4.1: The Impact of Configuration Diversity on Different Models

<table>
<thead>
<tr>
<th>Protocols</th>
<th>Configuration Diversity</th>
<th>SEIR</th>
<th>Four-Factor</th>
<th>STAWP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gnutella</td>
<td>0.3</td>
<td>0.5</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>GIA</td>
<td>&lt; 40 %</td>
<td>30-40%</td>
<td>70%</td>
<td>60-70%</td>
</tr>
</tbody>
</table>

Similarly, for a user behaviour value of 0.3, SEIR, Four-Factor and STAWP models show 30%, 40% and 90% infection densities for Gnutella protocol and 35%, 90% and 90% infection densities for GIA protocol respectively. It can be seen that, for the same values of a parameter, SEIR model shows the least infection density, compared to Four-Factor and STAWP models under Gnutella protocol. The reasons for this are same as those mentioned above. However, for GIA protocol, Four-Factor model demonstrates a very high value of infection. The reason is that Four-Factor model is not addressing the user-behaviour factor and the results are generated for a homogeneous network. GIA protocol also has a high query propagation rate due to its topology adaptation and biased random walk features. In the absence of any of these factors, the infection density appears to be very high for Four-Factor model considering GIA protocol.

### Table 4.2: Impact of User Behaviour on Different Models

<table>
<thead>
<tr>
<th>Protocols</th>
<th>User Behaviour</th>
<th>SEIR</th>
<th>Four-Factor</th>
<th>STAWP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gnutella</td>
<td>&lt; 30 %</td>
<td>50%</td>
<td>&lt; 40 %</td>
<td>50%</td>
</tr>
<tr>
<td>GIA</td>
<td>30-40%</td>
<td>30%</td>
<td>&gt;90%</td>
<td>80-90%</td>
</tr>
</tbody>
</table>
The results are generated for random values of configuration diversity and user-behaviour in the range (0.2-0.8) (Table 4.3). The reason for not choosing extreme values of parameters is because the worm propagation process stops if the worm come across an immunized node. The results presented in Table 5.3 show that SEIR model shows a 20% infection density for Gnutella protocol but a 30-35% infection density for GIA protocol. Similarly, for Four-Factor model, there is an infection density value of 35% for Gnutella protocol and of 55-60% for GIA protocol.

The parameter values are randomly produced in order to generate more realistic results. These results indicate approximately twice the infection population density value for GIA protocol, compared with that of Gnutella protocol. For a similar protocol, the infection density is twice for Four-Factor model as for SEIR model. This is due to the better query propagation and the biased random walk mechanisms implemented by GIA protocol.

<table>
<thead>
<tr>
<th>Table 4.3: The Impact of Random Values of Parameters on Different Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocols</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Gnutella</td>
</tr>
<tr>
<td>GIA</td>
</tr>
</tbody>
</table>

A homogeneous network is assumed in Table 4.4 for the generation of the results. It can be seen that the infection density is twice as much considering GIA as it is with Gnutella for SEIR or Four-Factor models. It can also be seen that the infection density is more than double for Four-Factor model, compared with SEIR model for both GIA and Gnutella protocols. The reason is that the assumption of a heterogeneous network is dropped and this causes the decrease in infection propagation.

<table>
<thead>
<tr>
<th>Table 4.4: The Impact of Random Values on a Homogeneous Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocols</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Gnutella</td>
</tr>
<tr>
<td>GIA</td>
</tr>
</tbody>
</table>
4.7. Performance Evaluation of SEIR Model using ABM Approach

4.7.1. The Concept of Behaviour Space

Analytical models have multiple settings with many input parameter values. All the parameters combined together are called parameter-space in mathematics and every output value has a particular combination of values. Running a model with different settings (and sometimes even the same ones) can lead to drastically different behaviour in the system being modelled. The question is how to produce a particular behaviour from the system with a given configuration of values.

Behaviour space in Netlogo is a better way to solve this problem. The results are recorded for different combinations of input values specified from the Netlogo interface. The behaviour of the system is dependent on the values of the GUI controls representing the parameter space. During the process, the model’s parameter space is sufficiently sampled so that the relationship between the different slider values and the behaviour of the system can be fully understood. During the execution, the output data can be recorded and visualized in in a pre-defined output format such as tables on spreadsheets.

4.7.2. Description of SEIR Model based on ODD Protocol

The ABM for SEIR analytical model, in terms of its purpose, state variables and scales, process overview and scheduling, design concepts, initialization, input and sub-models, is described below:

4.7.2.1. Purpose

The purpose of the active worm propagation SEIR model is to study the impact of worms on the network within the given time. This model addresses the configuration diversity of the network and the user behaviour of nodes as parameters that affect the worm propagation process. The simulation considers only unstructured P2P networks with particular consideration being given to Gnutella and GIA protocols [16].

4.7.2.2. State variables and scales

The model consists of the following three levels: individual, network and environment. The individuals are nodes that are operated by a user. The user applies user-specific settings such
as antivirus or antimalware programs, or setting an upload/download limit. It also includes P2P client on/off behaviour by the user. The infection time-lag is dependent on the configuration settings of the individual node and hence is the property of individual nodes as well. On the network level, configuration diversity and a scale free network (SFN) with a minimum and maximum degree of nodes are the important properties. The environment consists of overall infectious, exposed, susceptible and recovered nodes. The protocol (Gnutella or GIA) followed by the peers for worm propagation is also under consideration.

4.7.2.3. **Process overview and scheduling**

The model proceeds in discrete time-steps. With each time-step, a node can go from one state to another state from all the four defined states, namely, susceptible, exposed, infectious and recovered. The process starts from a node which is in an infectious state. The node can transit from a susceptible to an exposed state, which is a new state defined in the model to address the infection time-lag. The susceptible node can also transit from a susceptible to a recovered state if antivirus or other protection against the worm is available on the node. From the infectious state, the node goes to the recovered state, which is the absorbing state. The process overview and scheduling is shown below in Figure 4.16:
4.7.2.4. Design Concepts

- **Emergence**: active worm propagation dynamics emerge from the behaviour of the worm on an individual node, but the propagation is dependent on many factors.

- **Adaptation**: all the nodes keep a timer for the infection time lag when exposed. As soon as, the time reaches zero, the nodes switch from the exposed to the infectious state [143].

- **Fitness**: the nodes should be at least in one of the four states (susceptible, exposed, infectious and recovered).

- **Prediction**: the probability of neighbours being the next potential target of the worm is determined by computing the factors used in SEIR analytical model.
• **Sensing**: the node must be in an infectious state in order to transmit the infection to its immediate neighbours. The modelling of each state and of the state-transition is explicitly performed.

• **Interaction**: nodes communicate with each other following Gnutella or GIA protocol.

• **Stochasticity**: the user behaviour and configuration diversity are the factors that introduce stochasticity into the model. The values of user behaviour and configuration diversity are determined from the probability density function generated using Gaussian distribution with mean and standard deviation values.

• **Collectiveness**: there is no grouping of individual agents. All nodes are spatially independent and purely distributed and connected using P2P scale free networks.

• **Observation**: for model testing, spatial distribution of individual nodes is not required.

**4.7.2.5. Initialization**

The model can be initialized from 1 to 10 initially infectious nodes. The nodes are randomly selected from the whole network and the number can be predefined using the GUI of the model.

**4.7.2.6. Input**

In general, the configuration diversity and user-behaviour are determined from the probabilistic distribution. The total number of nodes from which to generate the network is specified. Other inputs, such as the infection time-lag, are provided. The number of neighbours of each suspected node and the selection of its next target among the neighbours is done based on values computed by the analytical model.

**4.7.2.7. Sub-models**

There is not sub-model involved in modelling the analytical model.

**4.7.3. SEIR: Simulation Setup**

The analytical model is implemented in Netlogo based on Gnutella and GIA protocols. The parameters such as the configuration diversity, user behaviour, infection time-lag, state-transition rate, node capacity, and minimum and maximum number of neighbours could be specified from the GUI of the Netlogo implementation. The results are generated using a
behaviour-space tool provided by Netlogo and from multiple simulated executions of the program.

A brief description of the different parameters used in the analytical model and implemented in the modelling follows:

1- Number-of-node: the number of nodes in the unstructured P2P network.
2- Infection-time-lag: the time needed for a node to be infectious or to remain an exposed state. It is dependent on the number of packages installed and the presence of anti-virus or anti-malware on the machine.
3- Initial-breakout-size: the number of initially infectious nodes before the start of the worm propagation process.
4- Diversity: the configuration diversity value of each node.
5- Behaviour: the behaviour of the user computed within the range of 0 to 1: zero represents a no user action while one represents a maximum user actions with a P2P client that can stop the worm propagation.
6- Average token: this parameter is used to represent the capacity of a node to handle the request. The greater the number of tokens, the higher will be the capacity of the nodes. This term is specific to GIA protocol.


A network of 500 nodes is generated during the simulation set-up. The number of nodes is limited because Netlogo provides visualization support of the process, which needs relatively more processing and memory capacity from the hardware perspective.

Initially, the values of different parameters, such as the configuration diversity and the user-behaviour of nodes, are set to zero to monitor the worm propagation process. However, the impact of these values with different variations is checked and validated with conventional modelling results. The graphical user interface of a sample set-up is shown in Figure 4.17 below.
4.8.1. The Impact of Configuration Diversity

Two benchmark models, Four-Factor and STAWP models, along with the proposed analytical model SEIR are considered and simulated for different values of configuration diversity in order to understand the worm propagation process. The impact of configuration diversity on worm propagation is shown in Figures (4.18) and (4.19) because both Four-Factor and SEIR models consider configuration diversity to be an important factor while STAWP model ignores it.

The results are generated for a configuration diversity value of 0.3 and 0.5. Similar to conventional modelling tool results (Figure 4.5), the implementation using ABM shows that SEIR model shows the lowest infection density among all the models. Comparing it with Four-Factor model (Figure 4.18(b)), there is a minor delay (Figure 4.18 (a)) in infection density for SEIR model. The reasons are an additional exposed state and the number of nodes that are recovered from the exposed state. STAWP model achieves a very high infection propagation within a very short period of time. The reason is that it does not consider the configuration diversity factor.

Figure 4.17: Netlogo GUI for Implementation
Figure 4.18: Infection Propagation over Time for Gnutella with CD = 0.3
Figure 4.19: Infection Propagation over Time for Gnutella with CD=0.5

(a): Configuration Diversity = 0.5 (SEIR Model)

(b): Configuration Diversity = 0.5 (Four-Factor Model)

(c): Configuration Diversity = 0 (STAWP Model)
4.8.2. The Impact of User Behaviour

As demonstrated earlier, user behaviour describes the actions a user performs with a P2P client. Logically, these actions slow down the query propagation and hence the worm propagation process. A review of the actions, including the user-behaviour parameters, is provided in Section 4.3.2. The results, shown in Figures (4.20) and (4.21), are generated for different values of user behaviour for different benchmark models.

SEIR model shows a 20% infection diversity for the user-behaviour value of 0.3 while it shows more than 20% of infection density for the value of 0.5. The results represent the direct impact of user-behaviour on the worm propagation process.

STAWP and Four-Factor models remain unaffected from the variation in this parameter as these models do not address this particular parameter. These results agree with the results already obtained using conventional tools (Matlab) in Section 4.3.2.
Figure 4.20: Infection Propagation over Time for Gnutella with UB=0.3
Figure 4.21: Infection Propagation over Time for Gnutella with UB=0.5
4.8.3. The Impact of Random Values of Parameters

The factors under consideration, namely, configuration diversity and user-behaviour, are considered with random values within the range of 0 and 1. These values are the lower and upper limits that apply to these variables. It can be seen that the infection density is in the range from 20-25% for SEIR model.

The results provided in this section match with results in Section 4.3.3. However, there is an exception in the random and quick early-increase in infection density for Four-Factor and SEIR models. The major cause of this variation is the network topology and finding highly connected nodes at early stage of infection process leads to a high infection density in the worm propagation process.

In this section, GIA protocol is implemented with an ABM tool (Netlogo) and integrated with the proposed analytical model. The salient features of the implementation are as follows:

1- The size of a node is proportional to its capacity to handle requests. This feature is referred to as the topology adaptation [16].

2- The token based mechanism of GIA for handling requests is implemented. This feature is also called the Active-Flow Control [16].
3- A biased random walk instead of a random walk is implemented as part of GIA implementation. This is also a core feature of GIA protocol [16].

4- A network of 500 nodes is considered.

5- 20 executions of each variable setup to obtain a better understanding of the behaviour of the worm in the network.

6- Active worms are considered which are worms that are topology-aware and know active IPs in the network [144].

The impact of different values of configuration diversity and of user-behaviour is measured. The purpose of these implementations is to verify the effectiveness of Agent-Based Modelling in comparison with conventional implementations. The results generated in this section are compared and analysed with the results generated in Section 4.4.

4.9.1. The Impact of Configuration Diversity

The value of infection density is higher in GIA protocol. In Gnutella protocol, however, similar values for configuration diversity were observed for all analytical models. The results are shown in Figures 4.23 and 4.24.

While the infection density is overall higher than that in Gnutella protocol results, there are a few observations concerning the simulation. First, there is a delay in the worm propagation process for some units of time and followed by a sudden increase. The reason for this is the token-based active flow control mechanism. A node that has to forward the query request asks the query handler to serve its request. If the serving node is busy and does not have the capacity to fulfil the request, it holds the request for some time until its resources are released to be served to the waiting node. This is the reason for the delay. The second observation is the relatively higher infection density. The reason is that the high-capacity nodes have a higher number of neighbours and a biased-random walk which forwards the query requests to highly connected nodes. The same formula is applied to all the infection propagation processes, regardless of the values of the parameters. The third and important observation is the die-out behaviour of simulations: this is when the simulation stops quickly and the worm does not spread with significant infection density. A detailed discussion of this observation will be provided in the next section.
Figure 4.23: Infection Propagation over Time for GIA and CD=0.3
4.9.2. The Impact of User Behaviour

The impact of user behaviour is similar to the effect provided in Section 4.4.2. SEIR model shows relatively lower infection propagation for both values i.e. 0.3 and 0.5 (Figures 4.24(a) and 4.24(b)), while the other benchmark models have no impact on worm propagation for the variation of these values. These reasons and observations are same as provided in previous section while considering impact of configuration diversity on user behaviour.
Figure 4.25: Infection Propagation over Time for GIA and UB=0.3
4.9.3. The Impact of Random Values of Parameters

All the factors (user-behaviour and configuration diversity) are considered randomly for simulating the worm propagation process using GIA protocol for all analytical models. The trends are observed same as that were observed in Section 4.4.3.

It can also be seen that all the graphs provided in Sections 4.8 and 4.9 do not exhibit the same decline in infection density over time as was shown with the implementation of conventional tools. The reason is the low recovery rate of the infectious population, apparent in Netlogo, compared with that in Matlab.
Figure 4.27: Infection Propagation with Random Values
4.10. Analysis and Discussion

There are many dimensions to the results that have been generated in this chapter, using conventional and non-conventional modelling tools. One dimension is the comparative analysis of the proposed analytical model ‘SEIR’ with the benchmark models considering the different parameters. The second dimension is an evaluation of the conventional and non-conventional approaches followed by an analysis of the variations in the results generated using SEIR model with different parameter values.

The work was extended by providing an integrated implementation of SEIR model with P2P protocols such as GIA and Gnutella in Netlogo. We thus provided an integrated solution of P2P protocols and an analytical model (Figure 4.28).

Figure 4.28: Integrated SEIR Model
4.10.1. Comparative Analysis of SEIR with Benchmark Models

Considering the random values of parameters for Gnutella, (Figure 4.6) shows infection density values of 30%, 40% and 90% for SEIR, Four-Factor and STAWP models respectively. Similarly, for GIA protocol implementation (Figure 4.10), these values are 35%, 60% and 90% for SEIR, Four-Factor and STAWP models. The infection density reported by GIA, compared with Gnutella, is higher for a similar network with SEIR model. The reason for this difference is the relatively efficient query propagation by GIA, using its topology adaptation feature and its biased random walk towards high-capacity nodes.

4.10.1.1. Simulation Die-Out Behaviour

Most of the time, the simulations performed in the literature focus on the peak values of infection density during the worm propagation process. During the computation, the results from the different simulations are averaged out. It has been reported in the literature that the infection may quickly die out, even when the prediction models report very high values during the epidemic outbreak [145].

The simulation results obtained for 20 executions for each analytical model, running under Gnutella and GIA protocols with different parameter values, show that a large number of infection processes end up in the early stages of the simulation and do not pass the minimum defined threshold value. A quantitative analysis of different simulations is provided in Tables 4.5 and 4.6. The results indicate that 55% of the time the infection dies out at maximum. On average, for any protocol, 41.66% of times the infection dies out quickly. Another important observation is that no die-out simulation is observed but only in GIA protocol. The reason is the topology adaptation feature in GIA, which tends to connect the nodes to a high-capacity node. The high-capacity node thus not only handles the load of queries (token-based flow control) but also acts as a super-peer for query propagation.

4.10.1.2. Network Lockdown

The term ‘Network lockdown’ has been coined for the phenomenon when during infection propagation, the infectious nodes are isolated from the recovered nodes in such a way that they are not able to propagate the infection. Due to this phenomenon, susceptible nodes remain in a suspected state and cannot be infected by the infectious nodes. This phenomenon is
observed for all analytical models operating under different P2P protocols (Figure 4.20 (a), 4.21(a), 4.22(b), 4.23(a), 4.23(b), 4.25(a), 4.26(a)).

This indicates that the presence of factors, such as configuration diversity and user-behaviour, can lead to a situation where infection cannot be further propagated even in the presence of a susceptible population. So these factors contribute towards taming the outbreak by providing natural immunization against the worm intended to spread within the network.

**Table 4.5: Die-out Behaviour for Configuration Diversity**

<table>
<thead>
<tr>
<th>Model</th>
<th>Protocol</th>
<th>Value</th>
<th>Total simulations</th>
<th>Die out Simulation</th>
<th>Die out %</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEIR</td>
<td>Gnutella</td>
<td>0.3</td>
<td>20</td>
<td>11</td>
<td>55</td>
</tr>
<tr>
<td>SEIR</td>
<td>Gnutella</td>
<td>0.5</td>
<td>20</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>SEIR</td>
<td>GIA</td>
<td>0.3</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SEIR</td>
<td>GIA</td>
<td>0.5</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>STWAP</td>
<td>Gnutella</td>
<td>N/A</td>
<td>20</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>STWAP</td>
<td>Gnutella</td>
<td>N/A</td>
<td>20</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>STWAP</td>
<td>GIA</td>
<td>N/A</td>
<td>20</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>STWAP</td>
<td>GIA</td>
<td>N/A</td>
<td>20</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>FF</td>
<td>Gnutella</td>
<td>0.3</td>
<td>20</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>FF</td>
<td>Gnutella</td>
<td>0.5</td>
<td>20</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>FF</td>
<td>GIA</td>
<td>0.3</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FF</td>
<td>GIA</td>
<td>0.5</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table 4.6: Die-out Behaviour for User-Behaviour

<table>
<thead>
<tr>
<th>Model</th>
<th>Protocol</th>
<th>Value</th>
<th>Total simulations</th>
<th>Die out Simulation</th>
<th>Die out %</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEIR</td>
<td>Gnutella</td>
<td>0.3</td>
<td>20</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>SEIR</td>
<td>Gnutella</td>
<td>0.5</td>
<td>20</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>SEIR</td>
<td>GIA</td>
<td>0.3</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SEIR</td>
<td>GIA</td>
<td>0.5</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>STWAP</td>
<td>Gnutella</td>
<td>N/A</td>
<td>20</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>STWAP</td>
<td>Gnutella</td>
<td>N/A</td>
<td>20</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>STWAP</td>
<td>GIA</td>
<td>N/A</td>
<td>20</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>STWAP</td>
<td>GIA</td>
<td>N/A</td>
<td>20</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>FF</td>
<td>Gnutella</td>
<td>0.3</td>
<td>20</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>FF</td>
<td>Gnutella</td>
<td>0.5</td>
<td>20</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>FF</td>
<td>GIA</td>
<td>0.3</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FF</td>
<td>GIA</td>
<td>0.5</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### 4.10.2. Comparative Evaluation of Conventional and ABM Modelling Approaches

The focus of this section is on the effectiveness and utility of the ABM and of conventional tools. The comparative analysis of the different worm propagation models will not be considered. Although a comparative analysis of ABM and conventional tools has recently appeared in the literature [98], however, the in-depth analysis of the role of tools in modelling the worm propagation process gained less attention. The comparative graph was generated for Matlab and Netlogo. The number of nodes for both simulations is 500, Gnutella is P2P protocol, the configuration diversity is 0.3, the birth-rate is 0.01 and the removal-rate is 0.01. The simulations are executed on the same hardware platform on the Windows operating system (Figure 4.29).
Figure 4.29: Comparative Evaluation of ABM and Conventional Tools

The important observations are as follows:

1- Matlab provides a fine grained control over the implementation of analytical models by providing a rich set of Application Programming Interfaces (APIs), compared with Netlogo. The functions of Matlab are optimized for performance and can produce results for a larger number of nodes, whereas Netlogo is limited to a smaller number of nodes due to its additional tasks of visual display and GUI controls. Another reason is the backend language in which the Netlogo is implemented, namely, Java. Matlab is implemented in C++.

2- The visual output of Netlogo helps in understanding the details of the worm propagation process while Matlab generates final results, and the underlying process is hidden from the user. The output and results can be controlled by the rich graphical user interface (GUI) provided by Netlogo while Matlab provides a command level control.
3- The support for debugging is an important aspect which has been missed by Matlab. Netlogo provides command level debugging. This function is achieved in Matlab by using output messages.

4- The results obtained for different states in the comparative graph (Figure 4.29) show similar trends but with minor differences. The worm propagation process in Matlab takes a longer time to start the transition for all states while the transition starts immediately in Netlogo. The reason is that the unit time considered by Netlogo can range from a few micro-seconds to many seconds depending on the actual processing time whereas Matlab provides the details about states at any instant in time. This is the reason for the delay in infection density trends with Matlab, compared with Netlogo.

5- Infection density requires both tools but minor difference have been found in their values. The principal reason is the network topology, which is a runtime variable and varies for each simulation.

6- A tool to monitor the behaviour of any process in Netlogo is provided; it is called ‘Behaviour Space’. Using this tool, any number of simulations can be executed and the results are stored in well-known output formats (CSV, Spreadsheet). Furthermore, these results can be analysed quickly and help in understanding the working of the overall process.

7- The interaction of agents and its visualization not only helps in understanding the worm propagation process but has also led to the development of immunization strategies.

4.10.3. Variation of Different Parameter Values in SEIR Model

4.10.3.1. Configuration Diversity

The impact of configuration diversity under Gnutella protocol for all benchmark models using conventional tools is shown in Figures 4.4(a) and Figure 4.4(b). Similar results for different configuration diversity values using conventional tools for GIA protocol can be found in Figures 4.8 (a) and 4.8(b). The results produced from conventional tools tally with the results generated from the ABM tools considering both protocols, Gnutella (Figure 4.18) and GIA (Figure 4.22).

All the results verify the following points:
1- The infection density observed in SEIR model is lowest in the presence of diverse sets of nodes in a network. The additional impact of suppressing the infection density in SEIR model is due to new states, such as the exposed state and more state transition options from any state to the Recovered state.

2- The higher the configuration diversity, the lower the infection density will be over time. It also implies that relatively more diverse networks have natural immunity against infection from a worm.

3- The value of infection density is higher for implementation using GIA protocol, compared with Gnutella protocol, under the same conditions. This verifies the efficiency of GIA for query propagation, by implementing one-hop replication, active-flow control and a biased random walk.

4.10.3.2. User Behaviour

The impact of user-behaviour under Gnutella protocol for all benchmark models using conventional tools is shown in Figures 4.5(a) and 4.5(b). Similar results are shown for different user-behaviour values using the conventional tools for GIA protocol in Figures 4.9(a) and 4.9(b). The results generated from the conventional tools are similar to the results generated from ABM tools under both protocols, Gnutella (Figure 4.19) and GIA (Figure 4.23).

The findings are as follows:

1- The user-behaviour factor minimizes the worm propagation process. The more the user operates P2P application, the lower will be the chances of infection propagation.

2- The results of conventional and non-conventional tools for different values of user-behaviour are the same under similar conditions. Minor differences in the results are due to the scale-free network that is generated during the runtime.

3- The value of infection density is higher for implementation under GIA protocol, compared with Gnutella protocol under same conditions. This is due to the same phenomenon identified in the previous section.

4.10.3.3. Random Parameter Values

To achieve more realistic results, random values of configuration diversity and of user-behaviour are taken within a range of 0 to 1. The results of this implementation using
conventional tool for different benchmark models under Gnutella are shown in Figure (4.6) and under GIA in Figure (4.10). The results using the ABM tool (Netlogo) are given for Gnutella in Figure 4.2) and for GIA in Figure 4.26.

The findings can be summarized as follows:

1- A smooth but relatively higher infection density is observed for random parameter values (configuration diversity and user behaviour) using SEIR model in GIA simulations, compared with Gnutella-based simulations. The reason for this is the deficiency of node capacity to handle the request in Gnutella (the scalability problem). A node has to wait until the other node finds enough resources to serve the request.

2- The concept of handling super-peers mathematically is introduced in SEIR analytical model. For a network that has a skew-degree distribution, a sudden increase in infection density over time is observed. However, during the averaging out of the results from a large number of simulations, this effect was minimized, resulting in relatively smooth graphs. The reason is that the worm or query reaches a node which is highly connected and the query or worm is then spread very quickly.

4.11. Validation of Theoretical and Practical Bounds of SEIR model

The theoretical limits of SEIR model are achieved using the parameters values described in the table 4.7. The same limits are validated by computing infection propagation with time using simulation. The upper and lower limits of parameters used in SEIR model are provided in table 4.7 as follows:
Table 4.7: The Bounds of SEIR Model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Lower Limit</th>
<th>Upper Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\tau$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$b$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$d$</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Consider a network of nodes with $N = 500$. At $t=0$, number of susceptible nodes are 499 while exposed and recovered nodes are zero. Only one infected node is considered initially. Complete system of equations for SEIR model is given below. Apply lower limit to the model at $t = X$ where $X = 500$, results in $S(X) = -499$, $E(X) = 499$, $I(X) = 1$ and $R(X) = 0$. It indicates that all the nodes in susceptible state will transit to exposed state in a uniformly degree distribution network. However, since rate of transfer from exposed-to-infectious and exposed-to-recovered is zero, all of the nodes will remain in the exposed state at lower limits.

For upper limits, consider the value of $\varphi = 1.5$ and other values same as defined in Table 4.7.

At time $t(0)$, $S(0)=499$, $E(0) = 0$, $I(0) = 1$ and $R(0) = 0$. It indicates that susceptible nodes will go to recovered state and there is no infection density.

To validate the theoretical limits, experiments with lower and upper bounds of values are performed in Figure 4.30 (a) and Figure 4.30 (b) respectively as follows:
Confirming to the theoretical bounds, lower limits keep all susceptible population to exposed and then infectious state. For better visualization, the lower bounds of state transition values such as $\gamma$, $\tau$ and $\rho$ are relaxed for minimal rate. Similarly, upper bounds values indicate that all nodes transit from susceptible to recovered state. There is no infection propagation in the case of upper bound of parameter values. These practical results validate the theoretical limits of SEIR model.

### 4.12. Summary

This chapter discussed the implementation of SEIR model using conventional network modelling tools and the ABM tool. Moreover, the impact of a variation of different parameters was observed, analysed and discussed. There are many findings associated with this research. For instance, the query propagation is 40% to 90% more efficient in GIA as compared with Gnutella. The reasons for this are the active-flow control, the one-hop replication and the topology adaptation features of GIA. Moreover, the configuration diversity, the infection time-lag and the user-behaviour combined together to show less infection density, resulting in hence less query propagation, as compared with other benchmark models such as (STAWP and Four-Factor models), thus providing a more realistic picture of the whole worm propagation process.

Another factor is the effect of super-peers in scale-free networks with skew-degree distribution, which causes a sudden increase in the infection density during the worm propagation.
propagation process. The reason for this is that the degree is not evenly distributed and a worm on a high-capacity node can cause a sudden increase in infection density. Considering same parameter conditions, infection population density is 15% more in case of GIA protocol as compared with Gnutella protocols due to super-spread phenomena in GIA protocol.

Furthermore, the usage of the ABM tool (Netlogo) for modelling the worm propagation process provides a better understanding of about the under-observation process, as compared with conventional tools such as Matlab. The salient characteristics are visualization of the simulation, an interactive variation of the parameters using GUI controls to monitor the effect on the under-monitoring process, executing multiple iterations using the ‘Behaviour Space’ tool and exporting results to provide an appropriate choice for modelling. Moreover, agents are autonomous entities that that communicate with each other to make decisions. This option makes the ABM tool stronger, as compared with a conventional tool such as Matlab, which uses a static structure such as e.g. Classes. In addition, modelling using ABM tools is relatively easier than using conventional tools due to the support of built-in constructs for writing simulation code. Although, the conventional tools have strong API’s, yet they are still not as effective as ABM tools.

In this chapter an exhaustive comparative evaluation of SEIR model with different benchmark models was provided. While there are many findings from the research, there are still certain limitations; they are the following:

1. A simulation with a conventional tool (Matlab) restricts the use of APIs to those provided by the tool. Since Matlab is a closed-source tool, the underlying algorithms and implementations are hidden. Therefore, customized solutions and behaviours are difficult to achieve through Matlab.

2. The ABM tool (Netlogo) provides excellent support for the process visualization. However, this restricts the number of nodes that can be used in the simulation. Although no upper bound is provided by the simulator documentation, a network of 1000 nodes significantly slows down the process.

In the next chapter, an evaluation of SEIR model from the perspective of a real-world worm will be provided.
CHAPTER 5 : SEIR: REAL-WORLD EVALUATION

5.1. Introduction

In Chapter 3, an analytical model, SEIR, was proposed for modelling P2P worm propagation process. This model considers the different factors affecting the worm propagation process and assumes the guise of a hypothetical propagation model of active worms. Based on this behaviour, a comparative analysis of the results was performed with the existing benchmark models using conventional and ABM tools.

To address the issues that are related to the absence of datasets for evaluation of analytical models, the data set and related case studies of Conficker worm are selected. In this chapter, the proposed SEIR model is evaluated in comparison with benchmark models which utilize the same worm propagation dataset obtained from the same data source on the propagation of the Conficker worm.

5.2. Conficker Spreading Mechanism and Behaviour

The Conficker worm appears to be a dangerous worm that has infected over 3.6 million Windows operating system-based machines. Different variations of the Conficker worm are reported in the literature [146]. Conficker A only relies on MS08-067 vulnerability for its propagation, while Conficker-B has more depth and implements two additional strategies [147]. The behaviour of this worm is active because it exploits a bug in the Internet Information Services (IIS) of the Windows machine. Conficker follows the TCP protocol by targeting port 445 [148] and exploits a vulnerability in the Microsoft Remote Procedure Call (RPC) procedure announced in MS08-067 released on 23rd October 2008 [147].

Conficker, just like other worms, adopts several advanced propagation strategies to infect the suspected hosts [149]. In the former study, the spreading mechanisms are classified into the following two categories: (i) global spreading hosts; and (ii) local spreading hosts. The infection spreads globally through Conficker by using a function for generating a random IP address. Although, this method has a global scale of infection, in the presence of devices such
as Intrusion Detection and Prevention Systems (IDS/IPS) and Firewalls, an internal network has fewer chances of infection from external sources [149]. In local spreading, Conficker can spread through, infecting hosts in the same subnet, infecting hosts in the nearby networks [148] and infecting portable storage devices [150].

5.3. An Analysis of the Conficker Worm Dataset

The availability of a real data set showing infected nodes and the propagation behaviour of the worm is a real challenge. Much of the state-of-the-art research on active worm propagation is based on hypothetical assumptions [148] [151] [95]. These works rely on simulations or analytical models to demonstrate the behaviour of worms. The major reason for such an approach by the researchers is the unavailability of a real dataset that would enable the severity of the worms’ attack to be accurately estimated [148][152]. The development of such models for worm propagation could then lead to the development of detection strategies with a potential high false-positive rate [153].

The latest available dataset for the Conficker worm was obtained from the Centre for Applied Internet Data Analysis (CAIDA³). The CAIDA network is a telescope, which scans the internet traffic by monitoring a large set of unusable IP addresses. Since these IP addresses are specifically for monitoring purposes, no normal network traffic is expected for these IPs. A detailed review of such traffic can thus provide a good insight into worm behaviour and can lead to an understanding of the propagation process.

The data provided by CAIDA is in the form of a trace file with (.pcap) extension. Wireshark⁴ is a tool that supports the extension and loads the trace files. The load test was performed on a 64-bit Windows operating system with a Wireshark packet capture tool (64 bit). The computation hardware was a quad-core machine with 32 GB of memory. However, due to the massive size of the trace file (3 GB approximately), it was not possible to load the whole file at once. The trace file was then split up into multiple trace files showing the data captured from a minimum of 200 seconds to a maximum of 600 seconds.

---

³ www.caida.org
⁴ https://www.wireshark.org/
Random scanning by the Conficker worm-targeting TCP protocol and port 445 resulted in an enormous increase in unique IP addresses per hour, compared with normal traffic. The CAIDA data shows 3222 unique IP addresses per hour before the outbreak of Conficker on 20th November, 2008 from 1800 hours to 0200 hours while this number increased to more than 100,000 unique IP addresses per hour on 21st November 2008 till 1600 hours\(^5\). This indicates that normal traffic is only 3.3% of the total traffic in cases of a Conficker outbreak.

The following graphs have been plotted with the total number of IP addresses in the trace file. The plot of this data has been presented in Figure 5.1, 5.2 and 5.3. In these plots, the green points represent total number of packets captured per second while red points represent packet captured with TCP error. The IP addresses communicating by using the TCP port 445 are around 63% while TCP erroneous data range from 16% to 20% (Figures 5.1, 5.2, 5.3). This includes the packets infected from other worms as well. However, the upper bound of the traffic infected from Conficker is between 60-65% for all data packets. Considering the percentage of normal traffic, the infection density due to Conficker could ranges from 57 to 62%. Also, it is important to note that the maximum value of the infection density with which Conficker can affect the network traffic has been considered.

<table>
<thead>
<tr>
<th>Table 5.1: Maximum Infection Propagation by Conficker</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Addresses (millions)</td>
</tr>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5.2: Sample 1 from One Trace File of Conficker Worm</th>
</tr>
</thead>
</table>

\(^5\)https://www.caida.org/research/security/ms08-067/conficker.xml
Figure 5.1: Sample 1 from One Trace File of Conficker Worm

Figure 5.2: Sample 2 from One Trace File of Conficker Worm
From all the graphs generated from the data, the following important points have become evident:

1- The number of infected nodes in all graphs is less than 65%. The same finding is demonstrated in [139], [154], which shows that Conficker have a low killing rate and a long-term period of propagation.

2- A machine infected from Conficker-C demonstrates 4 to 30 connection attempts per hour in a controlled environment [125].

3- The infection starting-point is missing in the data. As a result, the initial growth of infection is not determined. However, once it has reached the maximum infection, the graphs remain stable from that point onwards. This finding is compatible with the work done by Weaver [125], showing that the infection rate of Conficker is constant.

4- The data demonstrate that a large percentage of IPs in a network are getting infected, which contradicts our claim in Chapter 3. The role of factors such as Network Address Translators (NAT) requires to be further researched to understand the actual infection density in the network [149].

5.4. Related Work

The first Conficker outbreak was reported in November 2009 and since then, nearly a million machines remain infected even though the appropriate remedial measure have been taken.
The year-wise spread of the worm in the various countries of the world is shown in Figure 5.4; the maximum spread can be seen in 2010-2011.

The researchers in [155] developed an analytical model for the propagation of Conficker. Their model consists of two parts: a logistic part that represents the growth of the worm and an exponential decay part. Moreover, two major issues with this model were identified. First, it does not consider the impact of different factors on the worm propagation process. Second, it does not provide any variation in the results with different parameter values. VEISV (vulnerable – exposed – infectious – secured – vulnerable) network worm attack model has been proposed for measuring security counter-measures for worm propagation [156]. This model is based on SEIR epidemic model and its results are based on the data set obtained from CAIDA. The study provided in this work revealed that only 6% of the total host is infected from Conficker with an average infection time-lag for a host being found to be 3.5 hours. During this time, the worm performs the following tasks: checking the Windows operation systems version, copying the malicious code to the root directory of the operating system with a random file name, opening the firewall to create backdoors and randomly generating IP addresses. Conficker is reported as hybrid worm in [148] and combines three different spreading strategies, namely, local probing, neighbourhood probing, and global probing. The results are based on data provided by CAIDA. The traffic data captured by this project provides a good view of the various abnormal behaviours on the internet. Conficker is identified as a hybrid epidemic where different modes of spreading (local, global and neighbourhood) in isolation do not lead to the successful representation of the epidemic outbreak.

There is a behaviour analysis of the Conficker-C worm on a heterogeneous network in [125]. A non-homogeneous model is built that represents the scan generation process. This model is developed for ‘infection in the wild’ rather for a sandbox environment. However, the knowledge from both sandbox and typical user activity is used to generate the results.
5.5. **SEIR Model Evaluation with Conficker Worm**

In this section, SEIR model is evaluated for its accuracy and fitness for predicting the Conficker behaviour. To achieve these objectives, the results are generated using configuration diversity and user-behaviour values according to state-of-the-art research on these factors. These results are further evaluated with benchmark models that has been developed using the same dataset.

Two benchmark models are considered for the comparative analysis with SEIR model results. These are VEISV model [156] and Hybrid model [148]. Both of these models do not address configuration diversity, infection time and user behaviour as parameters affecting the worm propagation process. A comparative evaluation of SEIR model with the benchmark models will be given, taking infection density values into account. The results of the different state-transitions are shown below for the Hybrid model (Figure 5.5) and for VEISV model (Figure 5.6) respectively.

![Figure 5.5: Conficker Trends for Four Countries](image)
5.5.1. Performance Evaluation of SEIR Model

This section evaluates the performance of SEIR model in comparison with the benchmark models that have been developed based on the Conficker propagation behaviour. Figures 5.5 and 5.6 represent the Conficker worm propagation behaviour based on Hybrid and VEISV models. Figure 5.5 represent three different states such as suspected, infected and recovered.
along with time. Similarly, figure 5.6 representing exposed and infectious states under different reproduction rates using VEISV model. The results indicate that higher the reproduction rate ($R_0$) during infection process, higher will be the number of exposed and infected peer density. Thus in order to achieve network stability in shorter period of time, the security countermeasures taken in Vulnerable (V) state are effective. The decrease in infection outbreak in both infected and exposed state is observed by reducing the reproduction rate [156].

It can be seen that both models endeavour to fit the model-generated pattern to the data pattern provided by CAIDA. In contrast to this approach, SEIR model has different parameters such as the infection time-lag, configuration diversity and user-behaviour. Realistic values for these factors are taken from an analysis of the literature and the graphs are plotted using SEIR analytical model. The peak infection density value computed by the Hybrid model is 11.11%. VEISV model, on the other hand, shows an infection density peak of around 14% with a reproduction rate of 0.013.

Different types of network topologies such as star topology, ring topologies and networks such as small world, preferential-attachment and binary tree networks are considered when computing configuration diversity through the Shannon index in [157]. This study indicates that the power law network has a similarity index of nodes up to 62-65%. Another study about Gnutella-based clients shows that nearly 75% of clients run the same operating system, which is installed with similar packages [115], and therefore, considering all the other parameters to be the same, their chances of infection are also equal. The study concludes that only 25% of clients are diverse, using different operating systems with other packages installed. Based on this argument, the value of configuration diversity remains at 25% and 35% in Figure 5.7(a) and Figure 5.7(b) instead of random numbers being used in order to achieve more realistic results.

On the other hand, user behaviour is modelled as a random value in different studies [158] [159]. However, these studies indicate that there are, on average, 30 connection attempts by a single machine in a network infected by Conficker [125]. For the results generation, we also restricted the value of user-behaviour to be driven from [125].
The results were generated using a Barabasi-Albert (BA) Scale-Free Network of 500 nodes. The minimum number for a neighbour of a node was one while the maximum number was 10. A random-scanning pattern was followed for simulations in Figure 5.7. Configuration diversity was assumed to be 0.25 and 0.35 while the user behaviour was computed as 30/3600 connection attempts per second. Only a constant number of nodes in the network was considered with birth- and death-rate being zero.

![Graphs](image)

(a) Configuration Diversity = 0.25  
(b) Configuration Diversity = 0.35

**Figure 5.7: Infection Propagation considering Different Values**

It can be noticed that the graph generated using SEIR model considering Gnutella protocol (Figure 5.7) mirrors the graph shown for the VEISV model in Figure 5.6. The infected population density observed by SEIR model is 21% (Figure 5.7(a)) and 19% (Figure 5.7(b)) while the VEISV model shows 14% and the Hybrid model shows 11.11%. The increase-and-decrease behaviour of infection during worm propagation is similar for SEIR and VEISV models while the Hybrid model shows relatively smooth increase in infection density.

It is important to note that all of the three models do not show the 63% continuous infection rate that was shown by the real data, as illustrated in Figures (5.1), (5.2) and (5.3). The high variation in infection ratios will be discussed in terms of the NAT protocol.

Furthermore, the results are obtained using the ABM tool (Netlogo) to observe the propagation of Conficker with configuration diversity and user-behaviour values are considered to be the same as those in the previous section. The graphs shown in Figures 5.7 and 5.8 are similar, showing average infection density values of 22%.
5.5.2. Statistical Evaluation of Dataset

The reliability of dataset computed during experimentation is measured using Cronbach’s Aplha test. SPSS is used as tool for the measurement. The reliability of data is dependent on following criteria:

<table>
<thead>
<tr>
<th>Range of Cronbach’s Alpha.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha \geq 0.9$</td>
<td>Excellent</td>
</tr>
<tr>
<td>$0.9 &gt; \alpha \geq 0.8$</td>
<td>Good</td>
</tr>
<tr>
<td>$0.8 &gt; \alpha \geq 0.7$</td>
<td>Acceptable</td>
</tr>
<tr>
<td>$0.7 &gt; \alpha \geq 0.6$</td>
<td>Questionable</td>
</tr>
<tr>
<td>$0.6 &gt; \alpha \geq 0.5$</td>
<td>Poor</td>
</tr>
<tr>
<td>$0.5 &lt; \alpha$</td>
<td>Unacceptable</td>
</tr>
</tbody>
</table>

Figure 5.8: Infection Propagation over Time
The computed value of Cronbach’s alpha for the infection dataset is 0.965 using SPSS tool shows that data is reliable.

To compute the other statistical results of data generated from simulation of SEIR model and Conficker data, following dataset is used:

<table>
<thead>
<tr>
<th>SEIR Max. Infection Data (%)</th>
<th>Conficker Max. Infection Data (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.6</td>
<td>24.1</td>
</tr>
<tr>
<td>25</td>
<td>19.8</td>
</tr>
<tr>
<td>28.2</td>
<td>22.4</td>
</tr>
<tr>
<td>25</td>
<td>25.8</td>
</tr>
<tr>
<td>26.8</td>
<td>21.7</td>
</tr>
<tr>
<td>25.8</td>
<td>18.8</td>
</tr>
<tr>
<td>27.4</td>
<td>23.1</td>
</tr>
<tr>
<td>22.8</td>
<td>24.9</td>
</tr>
<tr>
<td>24.6</td>
<td>23.6</td>
</tr>
<tr>
<td>24.8</td>
<td>25.7</td>
</tr>
<tr>
<td>23.6</td>
<td>24.1</td>
</tr>
<tr>
<td>25.8</td>
<td>26.4</td>
</tr>
<tr>
<td>22.8</td>
<td>22.3</td>
</tr>
</tbody>
</table>

Table 5.5: Statistical Evaluation of SEIR and Conficker Dataset

<table>
<thead>
<tr>
<th>Variance (SEIR)</th>
<th>Variance (Conficker)</th>
<th>P-Value &lt; 0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results</td>
<td>2.65</td>
<td>4.80</td>
</tr>
</tbody>
</table>

The P-value = 0.9999 which by convention means this difference is considered to be not statistically significant at P < 0.05. Also, the variance in SEIR data is observed to be 2.65 where it is 4.80 for Conficker real dataset.
5.6. Discussion

A comparative analysis pattern was used to evaluate the different analytical models, as described in [145]. The infection population density from SEIR model was found to be 21% as a result of simulations using conventional and ABM tools. The variation in the results from SEIR model and the VEISV and Hybrid models is shown in Figure 5.9.

![COMPARATIVE ANALYSIS](image)

**Figure 5.9: A Comparative Evaluation of SEIR with Real Data and Benchmark Models**

The salient observations on Figure 5.9 are discussed below:

1. The infection propagation appeared to be higher, compared with the VEISV and the Hybrid models. The reproduction rate from the exposed to the infectious states is considered to be 0.138 for SEIR and for the VEISV models. Hybrid, being a combination of different propagation behaviours, shows the least infection density, compared with SEIR and VEISV. This suggests that global spreading is relatively slower than local spreading because the hosts in the same network have the same set of software and hence no configuration diversity. However, for global spreading, it assumes hypothetical configuration diversity values.
2- Raw data obtained from CAIDA shows 63% of traffic to be on port 445. Filtering the data by extracting the normal traffic produced Conficker traffic as being 55-60%. According to a recent study [81], around 34%–45% of DSL networks are using one IP address to serve multiple hosts. Since most of the Conficker-infected subnets are DSL networks, this ratio can be applied to results obtained using SEIR model. Since it is difficult to estimate the correct number of IP addresses due to NAT, an estimation ranging from 34% to 45% is provided in [81]. Applying the average value on Conficker traffic yields 20.5% IPs to be infected, which is closer to SEIR value of 21%.

3- The results of infection density obtained using SEIR model closely matches to the resultant infection density obtained from the data after NAT filtering.

4- The results provided by both conventional and ABM tools yields the same results the considering same values of parameters for SEIR model. The results are evaluated for a limited number of nodes (500). The evaluation of the scalability of the model for a large network size is reserved as a topic for future work.

5- Only a subset of the data can be evaluated due to the processing capability limitations of the tool ‘Wireshark’. It is required to load the large data set provided by CAIDA and to perform in-depth analysis. It is also suggested to develop customized applications that are based on low-level programming languages such as C or C++ that can load large datasets and perform an efficient analysis on such a dataset.

5.7. **Summary**

This chapter evaluated the applicability of SEIR analytical model to the real-world worm ‘Conficker’. The proposed model was evaluated in comparison with the benchmark models VEISV and Hybrid that had been developed based on a similar data set. The infection density computed using SEIR model based on real world values of parameters is compared with the results of dataset after NAT estimation. These results not only indicated that the infection density values achieved from SEIR model are relatively more accurate but also validated the application of SEIR model to a real-world worm and clearly proved the accuracy of SEIR model when compared with benchmark models.
CHAPTER 6: A NOVEL M-SEIR MODEL: INVESTIGATING THE ROLE OF MOBILITY FACTOR

In previous chapters, a novel active worm propagation SEIR model for unstructured P2P networks was proposed. This model addressed the impact of configuration-diversity, user-behaviour and infection-time-lag factors on the worm propagation process. The model was not only inclusive but also even has the flexibility to accommodate other factors.

In this chapter, the mobility of nodes, worm size, and wireless-link bandwidth and their impact on the whole worm propagation process will be considered. The node mobility factor will be proposed using the Gauss-Markov mobility model that will itself be integrated with SEIR model. The integration of these mobility-related factor will introduce an extended model ‘M-SEIR’.

6.1. Introduction

The evolution of smart phones with rich operating systems, a large number of installed applications and enhanced ease-of-use have resulted in a major focus-shift in users from desktop computers to smartphone devices. As a result, the Android, Symbian and Windows mobile operating systems are now major targets for malware [13]. Studies have shown that more than 65% of the malware has targeted the Android operating system since 2011 [13]. The fundamental reasons for this increase in threats to smart phones can be summarized as follows [13]:

- The open source kernel of mobile operating systems (Android), allows the malware writer to acquire an adequate understanding of the platform.
- Technological advancements over time, strong market competition and low prices are the major reasons for the wide spread use of smart phones.
- The capabilities of smart phones in terms of processing have been increasing exponentially. This has provided more opportunities for malicious users to exploit the platform.
• Writing malicious software for smart phones is similar to that for desktop computers. This makes it particularly easy for malware writers to move from one such environment to another very similar one.

6.2. Related Work

Mobility is an important aspect which is related to mobile devices in P2P networks. A analytical models of P2P worms, such as SEIRS [160], Four-Factor [93] and STAWP [7] did not consider the impact of mobility on the worm propagation process.

The impact of node mobility was not paid significant attention in the literature concerned with P2P worm propagation modelling. A survey of worms in mobile P2P networks, however, is provided in [161], according to which there are now over two billion mobile phones on the planet today. Statistics show that 63% of smart phones use Symbian operating systems. There are over 400 kinds of worm with 700 kinds of variant targeting the Symbian OS [161]. The propagation vectors of today’s mobile platforms are Bluetooth, SMS/MMS messaging, and situation applications, which are all being targeted by attackers [79].

There are different issues related to the node mobility that can affect the worm propagation process. One of these is the change in the topology of the interacting mobile nodes over time as the nodes move around in the environment. Secondly, the work done in the literature related to mobility considers short-range radio frequencies (RF), such as Bluetooth. There are other communication channels with higher bandwidths and data rates, such as WiFi, GSM, 3G and 4G that also need to be considered. In addition, the location of mobile devices in particular areas creates a certain density distribution, which itself can affect the worm propagation [122]. Another issue associated with mobility is that the worm spread is not studied in an environment consisting of wired and wireless nodes and in which the user is switching to different networks at different locations in an environment.

To examine the issues concerning wireless P2P networks, an algorithmic approach has been used for this can show the continuous availability of resources to mobile users throughout the wireless network environment [162]. A Mobility-aware file discovery control (MAFDC) scheme is used to update the peer status for better query propagation. However, the experiments are based on an NS-2 simulator with 802.11b as the selected protocol. This study
does not provide any information about other wireless networks, such as Wi-Max, Cellular, MANETs and wireless ad hoc networks. The authors in [163] developed an analytical model based on the SI epidemiological model. Their model considered the spatial dynamics of Mobile Ad-hoc Networks (MANETs), assuming that the worms are channel-aware with no mobility. In this model, the configuration diversity and the infection time-lag of nodes are not considered. The behaviour of worms on a low bandwidth, contention-full wireless channel with mobile nodes, compared to high-speed wired internet, is another issue that needs to be addressed.

The spread of a Cabir like malicious code via proximity-based, point-to-point wireless links has been studied in [145]. This work indicates that the existing worm propagation models provide hypothetical predictions and that, in fact, infection dies out early although these models show high infection values.

The study in [145] also discuss the reasons for the failure of worm- spreading models that do not consider node velocity and the configuration diversity of the network. Their proposed model, called “probabilistic queuing”, explicitly incorporates the ideas of connectivity skew and node mobility. However, their work does not consider the impact of interference and of the churn rate of nodes in the network on the worm-spreading process. The impact of worm-size on the wireless channel, using the capacity theory of wireless networks, has been provided in [164]. The results were generated for an SI epidemiological model for nodes moving in a wireless ad-hoc network. The results indicate that infection spreads linearly with the scale of network and uniform scanning strategy should be adopted to deal with the worm attacks.

A detailed description of worm propagation models for mobile networks is provided in Section 3.3.1.4.

6.3. Mobility Models in Networks

The mobility of nodes and the traffic communication patterns in an analytical model are important factors for modelling communication among mobile nodes. The movement of nodes, their location, velocity and acceleration are important parameters that need to be determined for the mobility of nodes. The objective of a mobility model is to emulate the object description as in a real-life. A list of the mobility models for Mobile Ad-hoc networks is shown in Figure 6.1 and overviewed below.
Random Movement-Based Mobility Models: This model is based on the random movement of nodes without any restriction. The mobility-related parameters of one node, such as direction, speed and destination, are also random and independent of other nodes. The Random Waypoint Model (RWP) is an example of these models. The RWP model is limited in terms of temporal dependency on the velocity of the node, the spatial dependency of the node and the geographic restriction on the movement of the node.

Temporal Dependence-Based Mobility Models: The movement information and patterns of a mobile node are affected by physical laws of acceleration, velocity and direction. The current speed and direction of a node are dependent on previous speed and direction. Therefore, the speeds and directions of a node at different time slots are correlated, which is called ‘temporal dependency’. The Gauss-Markov (GM) mobility model is an example of such a model where the velocity of the mobile node is assumed to be correlated over time and modelled according to the Gauss-Markov stochastic process.

Since most of the work in P2P networks makes use of these two types of model [166][164], a brief overview of the usage of these models in the context of P2P networks is now given. A framework for modelling malware dynamic in P2P networks with a heterogeneous environment is presented in [122]. Their experiments were carried out using an OverSim simulator, and their results were obtained using both a Random Waypoint Model and a GM
Mobility model. A three-dimensional framework, based on three different sets of parameters, is formulated in [167]. The first set includes the number of peers, the moving speed of peers and movement models. The second set contains churn models and the third set consists of the query success rate, the query time and the network loads. Gnutella and GIA are considered as P2P protocols during the simulation using the OverSim simulator. The RWP model and the GM mobility models are used to assess the impact of mobility.

The mobility of the node is an important aspect that has not been covered in previous worm propagation models collectively with other factors. SEIR model, proposed in Chapter Three, has the flexibility to accommodate mobility as a factor in the worm propagation process and to demonstrate the cumulative results, based on user-behaviour, configuration diversity and mobility. In this research, the GM mobility model will be used to compute the mobility factor and integrate it with SEIR model.

6.4. Factors Affecting the Worm Propagation Process in Wireless Networks

In this section, the parameters related to mobile node communication on wireless channels are discussed. These parameters include the mobility of the node, the bandwidth of the wireless-link and the worm size. The idea of infection-spread probability from a mobile node to its connected immediate peers will also be briefly discussed.

6.4.1. The Gauss-Markov (GM) Mobility Model

The mobility of a node is dependent on velocity, acceleration and the rate of change of direction. The current velocity of a node may be dependent on its previous velocity. Thus, at different time slots, the velocities of a single node are ‘correlated’ and this is called the temporal dependency of velocity. The Gauss-Markov (GM) mobility model works with this temporal dependence of velocity.

Cellular/wireless nodes are placed inside the coverage area of the base station /access point, and the nodes in the coverage area are independent of each other. Speed $\bar{S}$ is the mean speed and direction $\bar{\Theta}$ is the mean direction. For any time $t$, the speed and direction of a node is computed based on time (t-1)’s speed and direction using equation (6.1) and (6.2):
\[ S(t) = \alpha S_{(t-1)} + (1 - \alpha) \overline{S} + \sqrt{(1 - \alpha)^2} S^G_{(t-1)} \] \hspace{1cm} (6.1)
\[ \theta(t) = \alpha \theta_{(t-1)} + (1 - \alpha) \overline{\theta} + \sqrt{(1 - \alpha)^2} \theta^G_{(t-1)} \] \hspace{1cm} (6.2)

Where \( \alpha \) is the randomness parameter, the degree of randomness decreases as the value of \( \alpha \) increases and vice-versa. When \( \alpha = 0 \), randomness is very high and may lead to sharper turns. When \( \alpha = 1 \), the previous \( S \) and \( \Theta \) are more important, it means the current \( S \) and \( \Theta \) are more temporally dependent on the previous values.

The term \( S^G_{(t-1)} \) and \( \theta^G_{(t-1)} \) are random variables chosen from the probability density function (PDF) generated using Gaussian distribution with mean=0 and standard deviation =1.

The value of coordinates at time \( t \) is:
\[ X_t = X_{t-1} + S_t \cos(\theta_t) \] \hspace{1cm} (6.3)
\[ Y_t = Y_{t-1} + S_t \sin(\theta_t) \] \hspace{1cm} (6.4)

The distance, \( d \), from the base station can be calculated as:
\[ d = \sqrt{(X_t - X_0)^2 + (Y_t - Y_0)^2} \] \hspace{1cm} (6.5)

Thus the mobility factor can be computed as \( \eta = d/r \), for all \( d \leq r \).

6.4.2. Bandwidth Factor

On wireless channels, the data transmission capacity is much less than that on a wired network. Bearing in mind this fact, the issues and security challenges faced by mobile nodes are the following:

- The bandwidth of wired links on the internet is sufficient to transmit large data packets or worms within the shortest time. In contrast, the transmission links of wireless nodes are shared and have low throughput capability.
- A wireless node can transmit data to other wireless nodes using two options [164]. The first is direct point-to-point transmission, which requires a large radius of a base-station and hence suffers from interference. The second option is multi-hop transmission. In either case, the throughput achieved on wireless networks is less than that on wired networks.
The calculation of the bandwidth factor, using the wireless link capacity and the worm size provided, is as follows:

1- The mobile nodes are considered to be connected to a fixed base station through a wireless connection. The base-station is assumed to be connected to the through wired links. The bandwidth of wired links is ‘C’ while the maximum bandwidth of wireless links is ‘W’ where W < C.

2- Consider ‘S’ as the size of worm.

3- Let ‘n’ be the number of wireless nodes connected to a base station.

4- Consider only one node can be served at a time for contents at time ‘t’ from the base station considering only one transmission channel.

5- If the throughput of wireless link is represented by \( \rho \) then the condition \( \rho < W \) applies.

6- The number of nodes that can be infected from one node during time ‘t’ are \( \frac{\rho t}{S} \).

   Similarly, for unit time t, the rate of infection for a network of ‘n’ nodes connected to a base station is \( \frac{\rho}{nS} \).

7- The throughput of a fixed wireless random network is calculated according to [168] is as follows:

\[
\rho = \left( \frac{W}{\text{sqrt}(n \log(n))} \right) \quad (6.6)
\]

The rate of infection ‘\( \psi \)’ in a fixed wireless network with ‘n’ nodes connected to one base station can be computed as follows:

\[
\psi = \left( \frac{W}{S \times \text{sqrt}(n^3 \log(n))} \right) \quad (6.7)
\]

Equation (6.7) is valid if and only if the number of mobile nodes, \( n > 0 \) and \( \psi = 0 \) if \( n = 0 \).

6.4.3. Infection Spread Probability by Mobile Node

Consider a mobile node moving away from the base-station/access-point with uniform speed ‘S’. If the node has just been infected from any of the other peers, what is the probability that it can infect other suspected immediate neighbours? The formulation of the problem is shown in Figure 6.2 below.
‘M’ is the mobile node moving with speed ‘S’ away from the base station or access-point. $d_G$ is the distance that the node has already covered. $d_L$ is the remaining distance of the node from the effective boundary of the base-station. ‘R’ is the radius of base-station while $S_M$ and $S_F$ are suspected mobile and fixed nodes respectively. If $T_{\text{leave}}$ is the time in which the node can leave the effective radius of the base-station if moving with uniform speed ‘S’ in the same direction, and $T_{\text{transfer}}$ is the worm transfer time from an infected to a suspected node, then it can be computed as:

$$T_{\text{leave}} = (R - d_G)/S \quad (6.8)$$

$$T_{\text{transfer}} = W/\psi \quad (6.9)$$

And infection probability ‘$P_I$’ can be computed as follows:

$$P_I = T_{\text{leave}} / T_{\text{transfer}} \quad (6.10)$$

Where ‘W’ is the worm size and ‘$\psi$’ is the infection rate for the wireless node computed in Section 6.4.2.

### 6.5. Integration of Mobility Factor with SEIR Model

In Chapter 3, SEIR analytical model for active worm propagation was proposed. In this section, the Gauss-Markov (GM) mobility model is integrated with SEIR model so that the...
impact of the mobility of nodes on the worm propagation process can be observed. A detailed
description of the computation of the mobility factor ($\eta$) is provided in Section 6.4.1. It is
assumed that the mobile node is connected to P2P network via the hotspot/base-station, which
has a predefined effective radius ‘r’. P2P network is considered heterogeneous, consisting of
static and mobile nodes. The ratio of both types of nodes varies at different instants of time.
At any given time ‘t’, all the nodes could be mobile or all the nodes could be static or it could
be a mixture of both.

The GM model operates in a discrete time fashion and with each time tick ‘t’, the mobility
factor of the mobile nodes is updated. The value of the mobility factor is updated with each
unit of time for a mobile node and affects the worm propagation process.

The mobility factor is assumed to be an independent variable and does not have any
relationship with any other parameter. State transitions are detailed in Section 3.5. The
notations used in M-SEIR model are provided in Table 6.1 below.
Table 6.1: Notations and Description

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Total number of nodes in the network</td>
</tr>
<tr>
<td>S(t)</td>
<td>Number of suspected nodes at time (t)</td>
</tr>
<tr>
<td>E(t)</td>
<td>Number of exposed nodes at time (t)</td>
</tr>
<tr>
<td>I(t)</td>
<td>Number of infectious nodes at time (t)</td>
</tr>
<tr>
<td>R(t)</td>
<td>Number of recovered nodes at time (t)</td>
</tr>
<tr>
<td>δ</td>
<td>Configuration diversity. $0 \leq \delta \leq 1$</td>
</tr>
<tr>
<td>β</td>
<td>User behaviour. $0 \leq \beta \leq 1$</td>
</tr>
<tr>
<td>K_i</td>
<td>Degree of node i</td>
</tr>
<tr>
<td>ε</td>
<td>Infection time lag</td>
</tr>
<tr>
<td>η</td>
<td>Node mobility</td>
</tr>
<tr>
<td>ψ</td>
<td>Bandwidth factor</td>
</tr>
<tr>
<td>ρ</td>
<td>Rate of transition from exposed to recovered state</td>
</tr>
<tr>
<td>γ</td>
<td>Infectious to removal rate</td>
</tr>
<tr>
<td>τ</td>
<td>Rate of transition from exposed to infectious state</td>
</tr>
<tr>
<td>b</td>
<td>Birth rate of nodes in the network</td>
</tr>
<tr>
<td>d</td>
<td>Death rate of nodes in the network</td>
</tr>
</tbody>
</table>

6.5.1. Model Assumptions

While all the assumptions of SEIR analytical model, developed in Section 3.5, remain valid, the following are additional assumptions, considering mobility as an additional factor in M-SEIR model.

The assumptions that are taken to develop M-SEIR model are the following:

1- Mean-speed: A seed value for random computation of mean speed (1-100 km/hour).
2- Mean-angle: A seed value for the random computation of the mean angle (0-360 degrees).
3- The Gaussian Factor: A factor used in the Gauss-Markov mobility model to compute the coordinates of the nodes.
4- Mobile-node: The number of mobile nodes in the network. Each mobile node is surrounded by a circle, which represents the effective radius of the base-station. Mobile nodes are initially considered, see Figure 6.3.

![Figure 6.3: Heterogeneous P2P Network](image)

### 6.5.2. Complete System

If \((\eta)\) is the mobility factor computed in Section 6.4, then SEIR model can be extended as follows:

\[
\begin{align*}
I(t+1) &= \tau E(t) - dI(t) - \gamma I(t) \\
R(t+1) &= \rho E(t) + \gamma I(t) + S(t) \left(I(t)\right)_{\delta}(\beta)(\eta) - dR(t) \\
E(t+1) &= S(t) \left(I(t)\right)_{\delta}(1 - \delta)(1 - \beta)(1 - \eta) - \tau E(t) - \rho E(t) - dE(t) \\
S(t+1) &= -S(t) \left(I(t)\right)_{\delta}(1 - \delta)(1 - \beta)(1 - \eta) + (\delta)(\beta)(\eta) + bN - dS(t)
\end{align*}
\]

(6.11)
Where \( S(t) + E(t) + I(t) + R(t) = N(t) \) if and only if birth rate of nodes in the network is equal to the death rate in the network i.e. \( b = d \). The factor ‘\( \eta \)’ is independent variable and increasing values in the range \([0,1]\) indicate more distant nodes from the base-station and thus reduce the worm propagation process.

### 6.6. Evaluation of M-SEIR Model

As we have already mentioned, the impact of the mobility of nodes on the worm propagation process has not been considered extensively. Most of the work achieved in this domain has been concerned with wireless technologies such as Bluetooth or GSM.

In this chapter two benchmark models were selected for the evaluation of the proposed M-SEIR mobility model. The first model was the SIM (Susceptible Infectious Mobile) [164] which addresses the spread of malware in a generalized wireless network infrastructure. The SIM model was developed from the conventional epidemic model (SI) and as a result of the application of the capacity theory of wireless networks[168]. This model uses uniform scanning during the worm propagation. The rationale behind this particular model is that wireless networks can communicate in two modes, namely, direct and multi-hop. For both of these modes, it is not possible to achieve a throughput equivalent to wired links. This model considers a wireless ad-hoc network, where each node can transmit a packet to all its neighbours within their one-hop transmission range. The second benchmark model was the Probabilistic Queuing Prediction (PQ) model [145], which demonstrates the behaviour of worms that spread via proximity-based, point-to-point wireless links. This model considers the node-velocity and non-homogeneous connectivity distributions and developed as a refinement of the SI epidemiological model.

In the next section, the evaluation of the proposed M-SEIR model will be carried out using both conventional and ABM tools.

#### 6.6.1. M-SEIR Model Evaluation using Conventional Tools

The results from M-SEIR model are provided, using conventional tools (Matlab) and ABM tool, (Netlogo) for mobility and bandwidth factors only. These results will be comparatively evaluated with those from the benchmark models i.e. SIM and PQ for infection propagation.
To assess the mobility of the node, the GM mobility model [169] was used to keep track of the previous values of the coordinates of the mobile nodes. The GM model introduces parameters to handle the random movement of nodes. The value of zero indicates a completely random movement, just as with the RWP model, while value \(0 < \alpha < 1\) defines the degree of randomness.

The parameters considered for computing the mobility factor in M-SEIR model are the following:

1. Radius = 5, the radius of the base-station or access-point
2. Alpha = 0.5, the parameter to handle the randomness of the node
3. Speed = 10, the speed of the object in kilometres per hour
4. Angle = 30
5. Mean speed = 10, in kilometres per hour
6. Mean angle = 30, the direction of movement of the mobile node
7. Mean = 0, the mean value for Gaussian distribution
8. Standard deviation = 1, the standard deviation to create Gaussian distribution

The random values for speed and angle are chosen from the Gaussian distribution, generated with Mean and Standard Deviation values. These values are used in the analytical model.

### 6.6.2. The Impact of Node Mobility Considering Gnutella Protocol

The selected benchmark papers provided only limited information about the mobility models and the details of implementation to achieve the provided results. Mobility as a factor in M-SEIR model considering Gnutella protocol, is shown in Figures 6.4 (a), 6.4 (b) and 6.4 (c). The mobility of a node at any instant of time is computed using the GM mobility model. The direction and speed for the GM model and the number of mobile nodes in the network are provided for the simulation. M-SEIR model assumes that mobile nodes are moving at the same speed and in the same direction. Other factors, such as configuration diversity and user-behaviour, are considered with different values (details about these factors can be found in Chapter 3). It is also important to note that the results were generated without taking the bandwidth factor into consideration. The reason for this was the inability of Matlab to handle the delay involved due to low throughput on the wireless link.
It can be seen that there is a decrease in the number of infectious nodes as soon as the mobile node moves away from the base-station and, as a result, the mobility factor increases. The reason for the decrease in the node infection rate is the distance of the node from the base-station. A distant node is affected by different factors, related to wireless transmission, such as a loss in signal strength, channel path fading and interference, which all cause a reduction in node throughput.

![Figure 6.4: Infection Propagation with Different Values considering Gnutella](image)

6.6.3. The Impact of Node Mobility Considering GIA Protocol

A decrease in infection propagation is observed with different mobility values, as shown in Figures 6.5(a) and 6.5(b) for M-SEIR model. It can also be seen that the value of infection propagation is higher, considering GIA protocol, compared to that for Gnutella protocol under
the same conditions. The reason for this is the biased random walk feature of GIA that directs traffic towards high capacity nodes whereas in Gnutella random flooding is applied.

Figure 6.5: Infection Propagation with Different Values considering GIA

6.6.4. Discussion

In this section, the impact of mobility on the worm propagation process is seen to be similar to the impact of configuration diversity and user-behaviour. The results concerning the status of the nodes in different states with different mobility factors are shown in Figure 6.6 below:
As shown in the Figures above, the density of infectious nodes decreases with increased values of the mobility factor in Figures 6.6 (a), 6.6 (b). A relatively higher value for infection propagation is observed in GIA, compared to that in Gnutella due to the topology-adaptation feature of GIA. Important observations relating to the mobility factor are:

1- The impact of high mobility values on infection propagation in different protocols, such as Gnutella and GIA, was observed. For when the mobility factor is 0.8, it can be seen in Figure 6.6 that the infection propagation is close to zero. Such a low value of infection propagation is due to the different phenomena that are related to the wireless
environment, such as a weak signal strength, the path fading effect and possible signal interference.

2- A low mobility factor shows high infection propagation values in GIA, compared with Gnutella, but both in the presence and absence of other factors. High infection propagation is due to the topology adaptation feature of GIA.

3- In the presence of additional factors, such as a heterogeneous network and user behaviour, the infection propagation is reduced further with a decrease in mobility factor values (See Figures 6.6 (c) and 6.6(d).

4- For a mobility factor value of 0.3, the infection propagation under GIA protocol is 33%, which is 7% more than that under Gnutella protocol (26%) (Figures 6.6(a), 6.6(b)).

6.7. **Evaluation of M-SEIR Model using ABM Tools**

Netlogo was used for modelling the impact in a network of the mobility of nodes on the worm propagation processes. A detailed description of the set-up, controls and processes related to Netlogo is provided in Section 2.7.

The benchmark models, SIM [164] and PQ [145] provide the infection propagation value under given variable values. The evaluation of the results from the SIM model was based on wireless ad-hoc networks, where nodes are moving randomly within a closed area. It was the posited that infection propagation increases as the mobility of the nodes increases. Very high infection propagation was demonstrated by the mobile nodes, using this model (Figure 6.5 (b)). Similarly, the PQ model showed relatively lower infection propagation (70%) with a specific churn rate of nodes from the network. This model provides useful corrections to the conventional Kephart-White Model [170], but it still does not take network heterogeneity and user-behaviour into account.

6.7.1. The Impact of Node Mobility Process Considering Gnutella Protocol

The graphs generated for M-SEIR model were for 500 node simulations with 20 mobile nodes moving in different base-stations with a radius of 20 units. The worm size was assumed to be 1 Mb and the wireless link bandwidth was assumed to be 1 Mbps.
Performance evaluations of M-SEIR model with both the PQ and the SIM models is given below:

1- The infection propagation observed in the case of M-SEIR model is relatively low (20%) (See Figure 6.7, as compared with that in the SIM model (100%) (Figure 6.8) and in the PQ model (70%) (Figure 6.9). The major reasons for this are the holistic consideration of configuration diversity, user behaviour, the infection time-lag, mobility and bandwidth factors in contributing up to a significant extent to the reduction of the infection propagation.

2- The size of worm plays a major role in infection propagation on wireless links. The impact is computed as a bandwidth factor in equation (6.7). During simulation, the process slowed down significantly where the worm had to be transferred to the wireless nodes. This leads to the conclusion that the worm propagation process in wireless networks is many orders of magnitude slower than that for wired networks.

3- The SIM model shows 100% infection propagation within a very short period of time, whereas M-SEIR model demonstrated the huge delays in infection spreading (12 minutes for 20 simulations) for a given worm size and link bandwidth for wireless nodes. The delay is realistic because the throughput observed by worms on wireless links is much less than that on wired links. Hence, the infection process slows down.

4- The PQ predication model shows 70% infection propagation for a given churn rate of nodes. It indicates that the infection propagation varies according to the varying values of affecting factors. Multiple factors introduced in M-SEIR model holistically showed relatively less infection propagation, compared with that in the PQ model.

5- For the same infection time-lag value, M-SEIR demonstrates that some nodes are getting infected while some are recovered as well (Figure 6.7). This represent a true heterogeneous network where different devices have different operating systems, packages, antivirus and anti-malware installed and it is less likely that 100% of the network gets infected. Considering a heterogeneous P2P network, the chances of getting infection propagation values of 100% (SIM model) and 70% (PQ model) are low.
Figure 6.7: Impact of Node Mobility of Infection Propagation (M-SEIR Model)

Figure 6.8: Impact of node mobility of infection propagation (SIM model) [164]
6.7.2. The Impact of Node Mobility on Worm Propagation Considering GIA

The results shown in Figure 6.7 comply with the results taken for conventional tools considering same values for mobility. The infection propagation peak value considering GIA protocol observed in (Figure 6.10) is lower than that observed for Gnutella protocol (Figure 6.7). The reasons are link-level delay by wireless nodes which occupy a slot on super-nodes and do not release it until the worm transfer completed.

Long-delays in infection propagation are observed in the case of simulations considering GIA protocol. Main reasons include the capacity of super-peer nodes and wireless link-delay which pose hindrance in infection propagation. So the recovered nodes occupy the suspected nodes early. Hence the infection propagation slows down.
6.8. Mathematical Analysis of M-SEIR Model

The mathematical analysis of M-SEIR model evaluates the role of different parameters in the model and discusses the impact of the exclusion of a parameter from the model on worm propagation process. The parameters that are considered for the discussion are configuration diversity, user-behaviour, node mobility and degree distribution of nodes in the network. The system of equations, represented in equation 6.11, is considered as the benchmark for the analysis.

6.8.1. Configuration Diversity

The configuration diversity of nodes in network is represented by ‘δ’. Excluding parameter δ from equation 6.11 produced equation 6.12 as follows:

\[
\begin{align*}
I(t+1) &= \tau E(t) - dI(t) - \gamma I(t) \\
R(t+1) &= \rho E(t) + \gamma I(t) + S(t) (I(t))^\phi (\beta)(\eta) - dR(t) \\
E(t+1) &= S(t) (I(t))^\phi (1 - \beta)(1 - \eta) - \tau E(t) - \rho E(t) - dE(t) \\
S(t+1) &= -S(t) (I(t))^\phi [(1 - \beta)(1 - \eta) + (\beta)(\eta)] + bN - dS(t)
\end{align*}
\]  
(6.12)

Since the value of ‘δ’ range from zero to one, any value of the factor ‘1-δ’ decreases the number of exposed node and hence the infectious nodes. However, for extreme value of δ = 1 completely stops the infection process and all susceptible nodes have to go in recovered state.

Figure 6.10: Infection Propagation considering GIA Protocol (M-SEIR Model)
state. It means that the configuration diversity of nodes in networks decrease the infection propagation. Excluding configuration diversity factor from the equation results in speed up of worm propagation process.

6.8.2. User Behaviour

User-behaviour is represented by ‘β’ in the system. Excluding the parameter from equation 6.11 produced equation 6.13 as follows:

\[
\begin{aligned}
I(t + 1) &= \tau E(t) - dI(t) - \gamma I(t) \\
R(t + 1) &= \rho E(t) + \gamma I(t) + S(t)(I(t))^{\phi}(\delta)(\eta) - dR(t) \\
S(t + 1) &= -S(t)(I(t))^{\phi}[(1 - \delta)(1 - \eta) + \delta\eta] + bN - dS(t) \\
E(t + 1) &= S(t)(I(t))^{\phi}(1 - \delta)(1 - \eta) - \tau E(t) - \rho E(t) - dE(t)
\end{aligned}
\] (6.13)

The system of the equation is representing the worm propagation process in a scale-free heterogeneous network of static and mobile nodes without considering the impact of user-behaviour. Since user-behaviour is considered as an independent variable with a value ranging from zero to one, removing the variable with an average value also reduces the number of exposed nodes and hence the number of infectious nodes as well.

6.8.3. Node Mobility

The mobility of nodes plays an important role in the worm propagation process and it is represented as ‘η’ in equation 6.11. Excluding the mobility factor produces the following set of equations.

\[
\begin{aligned}
I(t + 1) &= \tau E(t) - dI(t) - \gamma I(t) \\
R(t + 1) &= \rho E(t) + \gamma I(t) + S(t)(I(t))^{\phi}(\delta)(\beta) - dR(t) \\
S(t + 1) &= -S(t)(I(t))^{\phi}[(1 - \delta)(1 - \beta) + \delta\beta] + bN - dS(t) \\
E(t + 1) &= S(t)(I(t))^{\phi}(1 - \delta)(1 - \beta) - \tau E(t) - \rho E(t) - dE(t)
\end{aligned}
\] (6.14)

Equation 6.14 represents a static scale-free and heterogeneous network of nodes without mobility. Since the mobility factor is computed using different node-related parameters, it is considered as an independent variable. The value of the mobility factor ranges from zero to one, thus considering the average value reduces the number of exposed nodes and therefore, the infectious nodes.
6.8.4. Network Degree Distribution

The network degree distribution is represented by ‘φ’ where φ = 1 refers to a network with uniform degree distribution while the scale-free networks, considered in the worm propagation process, have skew degree distribution with φ > 1.

\[
\begin{align*}
I(t+1) &= \tau E(t) - dI(t) - \gamma I(t) \\
R(t+1) &= \rho E(t) + \gamma I(t) + S(t)(I(t)) (\delta)(\beta)(\eta) - dR(t) \\
S(t+1) &= -S(t)I(t) [(1-\delta)(1-\beta)(1-\eta) + \delta\beta\eta] + bN - dS(t) \\
E(t+1) &= S(t)I(t) (1-\delta)(1-\beta)(1-\eta) - \tau E(t) - \rho E(t) - dE(t)
\end{align*}
\]

(6.15)

The system of equations represents a network with nodes having even degree distribution. The process of infection is uniform and there are no super-spreading phenomena involve.

6.9. Summary

The role of mobility-related factors in the early phases of the worm propagation process was studied in this chapter. The impact of the mobility of nodes, the worm size and of the wireless-link bandwidth was observed in the results by considering different P2P protocols such as Gnutella and GIA protocols.

A decrease in the infection peak due to the increased value of node mobility was observed for both Gnutella and GIA protocols. These results led to the conclusion that node mobility reduces the infection rate due to wireless-environment-specific parameters, such as signal strength, path fading and interference.

Compared to wired links, wireless links significantly reduce the throughput, and due to this phenomenon; the worm propagation slows down significantly, compared with the results shown by the SIM and PQ models. Such a delay could not be shown in the simulations of M-SEIR model from Matlab and Netlogo due to the limitations of the tools. However, a delay in Netlogo simulations was recorded for Gnutella and GIA protocols (Netlogo can record ticks only in an observed mode). The delay was many times more than that observed for wired links. This suggests that infection propagation in wireless networks is slow, compared with that in wired networks. The reason for this is the high value of throughput on wired links, compared with wireless links for same value of bandwidth.
The results were tested for 500 nodes, which is admittedly a limited number of nodes, but this was due to the processing limitations imposed by the ABM tool (Netlogo). However, this model (M-SEIR) can be tested on a larger number of nodes, a possibility that can be realised in future work.

Bandwidth factor did not get significant research attention from the perspective of P2P worm propagation and demonstrated to have significant role during the process. It is observed that the process slows down many times in wireless networks as compared to wired networks.
CHAPTER 7 : CONCLUSION AND FUTURE WORK

7.1. Introduction

This chapter concludes the thesis and discusses potential future work. Firstly, there will be a brief but comprehensive summary of the research work carried out. Following this, a review of the objectives, which were initially set, will be given, and the methodology adopted to achieve these objectives will also be provided. This chapter identifies the limitations in the work and the potential directions for future work, indicated by the research carried out in this thesis.

7.2. Thesis Summary

This research primarily focused on the impact of worms in P2P networks. Throughout this thesis, P2P networks, issues in P2P networks, worms, types of worm and worm propagation modelling techniques were discussed. New holistically integrated worm propagation models in unstructured P2P networks were also proposed, taking different factors into account. Moreover, various issues associated with model implementation were discussed in detail.

As demonstrated throughout this thesis, P2P networks, its associated issues and challenges, worms, types of worm and worm propagation modelling techniques were discussed. Among the different threat and challenges in P2P domain, this research primarily focused on the impact of worms in P2P networks.

The research concludes that the worm propagation process is to a great extent affected by different factors such as configuration diversity, user-behaviour, the infection time lag and the mobility of nodes. Therefore, new holistically integrated worm propagation models in unstructured P2P networks were proposed, taking different factors into account. Moreover, several issues associated with model implementation were discussed in detail.

Furthermore, a comparative research study has shown that infection propagation can be reduced due to the natural immunity against worms that can be provided by a holistic
exploitation of the range of factors proposed in this work. Additionally, the exploitation of a promising modelling paradigm, Agent-based Modelling (ABM), in the P2P worm modelling context has been built and trialled in this research.

The thesis began by briefly describing P2P networks, security issues in P2P networks and worm propagation as a dominant issue in P2P networks. It further set out the thesis statement, and the aims, the objectives and the structure of the thesis; it provided a clear path that was to be followed in the subsequent chapters. Chapter Two, introduced P2P networks, the characteristics of P2P networks, types such as unstructured and structured P2P systems and security-related issues. Common unstructured P2P protocols, such as GIA and Gnutella, were described in detail. The chapter further elaborated on the issues and challenges associated with P2P networks, such as scalability, the free-rider problem, quality of service, node mobility, social networks, conventional modelling tools and worm threats. Different conventional modelling tools were discussed within the context of P2P networks.

Moreover, Agent-based modelling was introduced in this chapter as a new paradigm to solve the dilemma of the whole complex worm propagation process. Following this, multiple ABM tools were studied. The merits and demerits of ABM and conventional simulators were discussed from the perspective of P2P networks. A performance evaluation of GIA was then carried out, using conventional simulators such as Oversim and an ABM tool such as Netlogo. Hop-count and delay were considered as metrics of evaluation for both tools. It was found that ABM tools could be used to model different processes in P2P networks effectively. The chapter helped to lay the foundations for integrating the implementation of P2P protocols with the analytical models for worm propagation, using ABM tools.

A detailed insight into P2P worms, their propagation behaviour and their destruction strength by presenting existing analytical models was introduced in the third chapter. Various famous P2P worms such as Code Red, Nimda, Slammer, Blaster and Conficker were discussed in terms of their origin, the vulnerability they exploit and the impact they have had on the whole network. Moreover, a detailed discussion of worm propagation models, their classification and different common immunization strategies was provided. A classification of existing worm propagation models, based on the propagation behaviour of worms, was provided. Gaps in the existing research were identified in the form of parameters that were not discussed in the existing mathematical models holistically. These parameters included the infection time-lag,
the configuration diversity and the user behaviour, all of which can affect the worm propagation process. Consequently, a new stochastic worm propagation model (SEIR) for active worms considering all these parameters was developed.

Chapter four provided the results and a discussion related to SEIR analytical model, which was developed in Chapter 3. Two common unstructured P2P protocols, Gnutella and GIA, were selected as the protocols of observation for the results. The results were achieved using ABM tools and conventional tools. Furthermore, a detailed and comparative analysis of worm propagation behaviour was provided using different values for the factors with benchmark models. An integrated solution of SEIR model and of P2P protocol (Gnutella and GIA) implementation was found. The impact of different tools (ABM and Conventional) was evaluated from the perspective of the worm propagation process. It was concluded that the worm propagation process can be effectively represented by ABM tools and that this provided better controls, compared with the conventional network modelling tools. Certain aspects of the process, such as the simulation die-out behaviour and the network lock-down effect, which were not observed using conventional tools, were evident with ABM tools. It was also accepted that configuration diversity and user-behaviour can slow down the worm propagation process and can potentially serve as network immunizers.

The evaluation of the proposed SEIR model with a real-world data set for one of the most recent worms (Conficker) was carried out in chapter five. The data set was obtained from CAIDA and analysed using the Wireshark tool. The data showed the various trends of the Conficker-affected traffic. The infection density was realised using SEIR model and then compared with that realised with the benchmark models, using conventional and ABM tools. The infection propagation values provided by all of these models, including SEIR model, were compared with the results based on the infection density values provided by the real data-set. The evaluation demonstrated that the predictions shown by SEIR model relating to infection density in the network were closest to the infection density values produced by the real data-set.

Chapter six started with an investigation into the previous work achieved regarding the impact of the mobility of nodes during the worm propagation process. The study revealed that only limited work had been done in this regard and it did not reflect the true picture of the impact of node mobility on the worm propagation process. Following this, different mobility models,
such as the Gauss-Markov mobility model and the Random Way Point (RWP) model, were studied to understand the movement of mobile nodes. The Gauss-Markov mobility model was selected to calculate the mobility factor because it keeps track of node velocity, is flexible for adjusting the randomness degree and maintains the spatial-temporal behaviour of the mobile node. Similarly, the bandwidth factor was calculated, taking into consideration the various parameters of the wireless networks, such as link-bandwidth and worm size.

The role of mobility on P2P worm propagation was discussed thoroughly. Consequently, the mobility factor was integrated with SEIR model, which was thus extended to an M-SEIR model. The salient features of M-SEIR model are considering the spatial-temporal behaviour of nodes, node-level mobility-factor control, randomness control for individual nodes, throughput offered by the network and the worm size. The results were computed using Matlab and Netlogo, which again revealed that conventional tools fail to handle node-level control and the time-varying properties of the simulation.

7.3. Meeting the Objectives: Original Contributions

The thesis objectives were stated in the first chapter of this thesis. The main objective was to understand the worm propagation process and to develop an analytical model to achieve these objectives. In striving to realise these aims, several other objectives were met, the details of which are given below.

1. The first objective was to investigate different P2P networks, conventional network modelling tools and ABM tools. To achieve this first objective, a thorough analysis of different P2P networks and protocols was carried out and described in Chapter 2. Different network modelling tools (ABM and conventional) were studied in detail. A comparative analysis of both types of tools was done by selecting OverSim as the conventional tool and Netlogo as the ABM tool. The observer’s metrics were hop-count and delay. It was observed that ABM tools offered better help during the simulation by providing node level control, visual-aids and low-level programming support.

2. In Chapter 3, different worm propagation models, developed for unstructured P2P networks, were studied. These models were further classified, based on the propagation behaviour of worms. It was observed that most of the work done in the...
literature is related to active worm propagation modelling while relatively little work has been done for other types of worms. Different parameters were found that may affect the propagation process and a list of the parameters for each type of model was generated. It was also noted that existing worm propagation models consider only a sub-set of parameters and ignore the super-set, which itself could lead to more accurate results.

3. A stochastic worm propagation model (SEIR), considering the infection time-lag, user-behaviour and configuration diversity, was developed in Chapter 3. This model was evaluated for different parameter values, using infection density as the metric of measurement. The model was evaluated using Matlab and Netlogo. A major contribution was the implementation of two different P2P protocols, Gnutella and GIA, together with the analytical model. The results demonstrated that infection density is reduced in the case of high parameter values. Using ABM tools, important effects, such as simulation die-out behaviour and network lock-down effect, were observed. ABM was also shown to address worm propagation modelling better in terms of individual node handling, visual aid, the repetition of results for behaviour measurement, node coordination and handling varying time delays.

4. The evaluation of the proposed model (SEIR) was done, based on a real-world worm (Conficker) data-set obtained from CAIDA. The data-set was analysed using a Wireshark tool and Conficker-infected-traffic patterns were clearly observed. Following this, two different analytical models, selected as benchmarks, were given the same data-set. It was then observed that SEIR model matched closely the infection density behaviour provided by the real data, using the parameter values identified in the literature.

5. Addressing the impact of the mobility of nodes in an unstructured P2P network was set as a major objective. To find the answer, the mobility factor was computed for mobile nodes using the Gauss-Markov mobility model (Chapter 6). The impact of the wireless channel bandwidth and the worm size was also computed in the context of worm propagation. The mobility factor was integrated into SEIR model, thus expanding it into an M-SEIR model. The results from M-SEIR model were generated using Matlab and Netlogo and then comparatively evaluated.
7.4. Limitations

The research conducted in this thesis has provided an invaluable insight into understanding worm propagation behaviour. While important discoveries and contributions have been made by this research, there remain of course certain limitations. These are noted below.

- **Real-World Validation**: The overwhelming majority of work done in this research area is based on a hypothetical worm. In our work, we report an initial validation of our developed analytical model through a Conficker worm dataset. However, more extensive comparative analysis of real-world worms with additional real data is still required, including comparison with other state-of-the-art models.

- **Data Set Availability**: Limited access to worm data was provided by various organizations such as CAIDA and CRAWDED\(^6\). However, this data was filtered and was missing some important values. It was also hard to analyse a very large data-set with the available existing tools. Customized programs and scripts are therefore required to enable a better analysis of the data.

- **High Level of Abstraction**: The results provided in this research were generated using conventional and ABM tools. It was observed that most commonly used conventional tools such as Matlab and Oversim have only limited software development support and do not provide a fine-level of control on programs to observe specific P2P worm process behaviour. As demonstrated in this work, during simulations, these tools impose certain restrictions on modelling behaviour, such as having variable delays in the propagation process and having a customizable derivative with allotted time for infection propagation. Considering this, tools that provide a low-level of control during programming/simulations are urgently required.

- **Lack of Customized Tools**: Due to the lack of support of modelling most common P2P protocols in existing tools, it becomes difficult to evaluate the results generated by these tools. A basic implementation of the protocol along with a customization option is required to get valid results.

\(^6\) http://crawdad.org/
• **Mathematical Analysis:** Mathematical analysis of the proposed models has been discussed briefly. However, further mathematical evaluation still needs to be carried out of the impact of various state transition rates on the proposed models.

### 7.5. Future Work

This research work opens new avenues for further research. Some possible research directions are given below:

• **Hybrid Worms:** A worm that can adopt different propagation strategies at different times or under different circumstances in a context-aware manner, is called a hybrid worm. An analytical model that predicts the propagation behaviour of such hybrid worms is a subject that would warrant further research.

• **Big Data Analytics:** The model was evaluated with a small number of network nodes with average query propagation rates. Big data has recently been gaining attention as it is generated by a variety of devices, including the Internet of Things (IoT), medical imageries, credit cards, mobile device location specific data through GPS, video streams and social media applications. The impact of worms within the context of big data is another field that clearly needs to be investigated.

• **Scalability of the Analytical Model:** The existing analytical model (M-SEIR) was developed while considering all the relevant factors affecting the worm propagation process. However, the model is scalable and can be enhanced to accommodate propagation behaviour or relatively more complex worms such as hybrid worms. This is another interesting area for further research.

• **Synthetic Model:** The availability of a real data-set is a challenge for validating the results. Creating a more realistic data-synthesis model that learns and generates the propagation data of different real-world worms is another possible research direction.

• **Extension for Netlogo to Model P2P Worms:** Netlogo, an ABM tool, can be extended by incorporating different P2P protocols, such as GIA, Gnutella and Kaaza. This will allow researchers to effectively utilize ABM tools and more effectively simulate the worm propagation process in mainstream P2P networks.
• **Develop a New Immunization Strategy**: The parameters addressed in this research can help in devising immunization strategies for the impact of these parameters on immunization. Such strategies require further research and appropriate validation.
APPENDICES

Appendix I: M-SEIR Model and Biological Worms

In this section, a brief study of biological worms and their propagation behaviour is studied. An attempt is made to link the factors addressed in M-SEIR model with the causes of biological worm spread. Little tweaks in current M-SEIR model are suggested to model the propagation behaviour of biological worms.

A hookworm is affecting the humans from century from all over the world. The parasite of the disease get blood from intestine of infected hosts and spread via soil or other media [171]. It is observed that different persons have different immunity and therefore the infection probability from the hookworms is directly proportional to the immunity against worms. This refers to the configuration diversity factor of M-SEIR analytical model. It is also observed that the workers that are working in forests or constructions of rail road are less likely to be infected from hook worm as compared with sedentary working populations due to the reasons set [172].

The studies indicate that repeatedly visiting same infectious place increases the probability of infection via hookworms. The mobility factor is already presented in M-SEIR model, however requires little tweaks to map to the fact described. The impact of frequency of visits to the infectious location in hookworm transmission can be added in the current analytical model to address this factor.

Ebola epidemic began in Guinea during December 2013 and affected at least five countries. It spreads through different body fluids and can be stopped through early diagnoses, patient isolation, contact tracing and safe burial [173]. The incubation period, is the time between infection and the onset symptoms which refers to the infection time lag in M-SEIR model. A graph of infection over time for Ebola in West Africa is shown in Figure A.1.
Figure 8-1: Infection Propagation Over Time for Ebola

Figure 8-2: Infection with Time
Thelazia callipaeda (Spirurida, Thelaziidae) is known as the “oriental eyeworm” because of its occurrence in the Far Eastern Countries (i.e., Indonesia, Thailand, China, Korea, Myanmar, India and Japan). This nematode infection induces from mild to severe ocular manifestations in animals, as well as in humans [174]. The spread of this worm is effected by immunity of each individual, the time required to get infected and the location of the individual. These parameters can easily be mapped to M-SEIR model with little modification.

The aforementioned discussion shows that the proposed M-SEIR model can not only be used for modelling active P2P worms but can also simulate the behaviour of biological worms.
Appendix II: Age-Structure and Stage-Structured Solutions

The results from existing models for modelling the propagation behaviour of Conficker worm modelling are limited. The models do not address different factors such as infection time lag, configuration diversity and user-behaviour. An analytical model based on the concept of Lefkovitch matrix is created. Lefkovitch matrix is used for modelling state transition modelling in biological phenomena and handles the probabilities of transition [175], [176]. The Leslie matrix is another way of modelling age-structure of population.

This matrix is also called ‘population projection matrix’ and dependent on age-dependent fecundities and proportional survivorships. The Leslie matrix is the correct approach for animals, but not for many plants. For them size or stage is more important than age. Instead, plants (and things like corals) are better described using a transition matrix approach.

Taking the lead from this point, Lefkovitch is a transition matrix that can be used in the development of stage-structured analytical model. It contains all stage-specific transitions and seed production. There are i columns and j (= i) rows for a population with i stages as shown in table AII.1. The i=j, the value indicates that the fraction of organism that remains in the same state.

<table>
<thead>
<tr>
<th>A</th>
<th>This year (t)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stage 1</td>
</tr>
<tr>
<td>t+1</td>
<td></td>
</tr>
<tr>
<td>Stage 1</td>
<td>a_{11}</td>
</tr>
<tr>
<td>Stage 2</td>
<td>a_{21}</td>
</tr>
<tr>
<td>Stage j</td>
<td>a_{j1}</td>
</tr>
</tbody>
</table>

SEIR analytical model developed in chapter 3 is represented in the form of Lefkovitch matrix while keeping the same assumption as defined in chapter 3. Addition of different values of parameters and their impact on the worm propagation process will be observed in detail.

Analytical Model for Conficker Worm

The analytical model represented in this section is the continuation of SEIR analytical model developed in Chapter 4. Following assumptions and constraints are considered for the development of the analytical model:
1- The analytical model considers random scanning of IP addresses which is a feature of Conficker worm.

2- Non-homogenous network with diverse user behaviour is considered.

3- Both propagation modes, global and local, are considered in the model by considering the degree distribution of connected nodes.

4- The variables used in the analytical model are shown in table 8-2:

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Total number of nodes in the network</td>
</tr>
<tr>
<td>S(t)</td>
<td>Number of suspected nodes at time (t)</td>
</tr>
<tr>
<td>E(t)</td>
<td>Number of exposed nodes at time (t)</td>
</tr>
<tr>
<td>I(t)</td>
<td>Number of infectious nodes at time (t)</td>
</tr>
<tr>
<td>R(t)</td>
<td>Number of recovered nodes at time (t)</td>
</tr>
<tr>
<td>Δ</td>
<td>Configuration diversity. $0 \leq \delta \leq 1$</td>
</tr>
<tr>
<td>B</td>
<td>User behaviour. $0 \leq \beta \leq 1$</td>
</tr>
<tr>
<td>$K_i$</td>
<td>Degree of node i</td>
</tr>
<tr>
<td>E</td>
<td>Infection time lag</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>Infectious to removal rate</td>
</tr>
<tr>
<td>T</td>
<td>Rate of transition from exposed to infectious state</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>Birth rate of nodes in the network</td>
</tr>
<tr>
<td>$\nu$</td>
<td>Death rate of nodes in the network</td>
</tr>
<tr>
<td>$\Gamma'$</td>
<td>Exposed to infection state transition probability</td>
</tr>
</tbody>
</table>

Considering four states in the model, the state transition diagram is shown in figure 8-3:
Figure 8-3: State Transition Diagram

The transition probabilities are explained without considering the birth and death rates of a state as:

Table 8-3: Stage transition probabilities

<table>
<thead>
<tr>
<th>Transitions</th>
<th>Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>A21</td>
<td>((1 - \delta) x (1 - \beta ))</td>
</tr>
<tr>
<td>A41</td>
<td>((\delta) x (\beta ))</td>
</tr>
<tr>
<td>A32</td>
<td>(\pi)</td>
</tr>
<tr>
<td>A42</td>
<td>(1 - \pi)</td>
</tr>
<tr>
<td>A43</td>
<td>(\gamma)</td>
</tr>
</tbody>
</table>

Given the above transitions and their probabilities, the Lefkovitch representation of four-state analytical model is given as:

\[
\begin{bmatrix}
S \\
E \\
I \\
R
\end{bmatrix} =
\begin{bmatrix}
0 & -A21 & 0 & -A41 \\
A21 & 0 & -A32 & -A42 \\
0 & A32 & 0 & -A34 \\
A41 & A42 & A43 & 0
\end{bmatrix}
\begin{bmatrix}
S \\
E \\
I \\
R
\end{bmatrix}
\]

\[ \text{------- (A.1)} \]

The matrix elements with negative values representing the state leaving probabilities. Solving the matrices in equation (A.1) yields following system of equations:

\[ S \text{ (t+1)} = -A21 \times E \text{ (t)} - A41 \times R \text{ (t)} \] \[ \text{-----------------(A.2)} \]

\[ E \text{ (t+1)} = A21 \times S \text{ (t)} - A32 \times I \text{ (t)} - A42 \times R(t) \] \[ \text{------- (A.3)} \]

\[ I \text{ (t+1)} = A32 \times E \text{ (t)} - A34 \times R \text{ (t)} \] \[ \text{----------------------(A.4)} \]

\[ R \text{ (t+1)} = A41 \times S \text{ (t)} + A42 \times E \text{ (t)} + A43 \times I(t) \] \[ \text{---------- (A.5)} \]
This equations (A2-A5) represent exactly SEIR model developed in chapter 3. Lefkovitch matrix provides a way to develop analytical models for stage-structured population with defined transition probabilities. In the subsequent sections, the results using SEIR analytical model will be taken using parameter values obtained from literature and compared with benchmark models developed on the same dataset.
7. REFERENCES


200 | P a g e


