

**AN ELAPSED-TIME BASED
SCHEME FOR DETECTING AND MITIGATING
DDOS ATTACKS IN THE SDN ENVIRONMENT**

WISAM H A. MURAGAA

UNIVERSITI SAINS ISLAM MALAYSIA

**AN ELAPSED-TIME BASED
SCHEME FOR DETECTING AND MITIGATING
DDOS ATTACKS IN THE SDN ENVIRONMENT**

Wisam H A. Muragaa

Thesis submitted in fulfillment for the degree of
DOCTOR OF PHILOSOPHY IN
ENGINEERING AND BUILT ENVIRONMENT

UNIVERSITI SAINS ISLAM MALAYSIA

July 2019

AUTHOR DECLARATION

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged

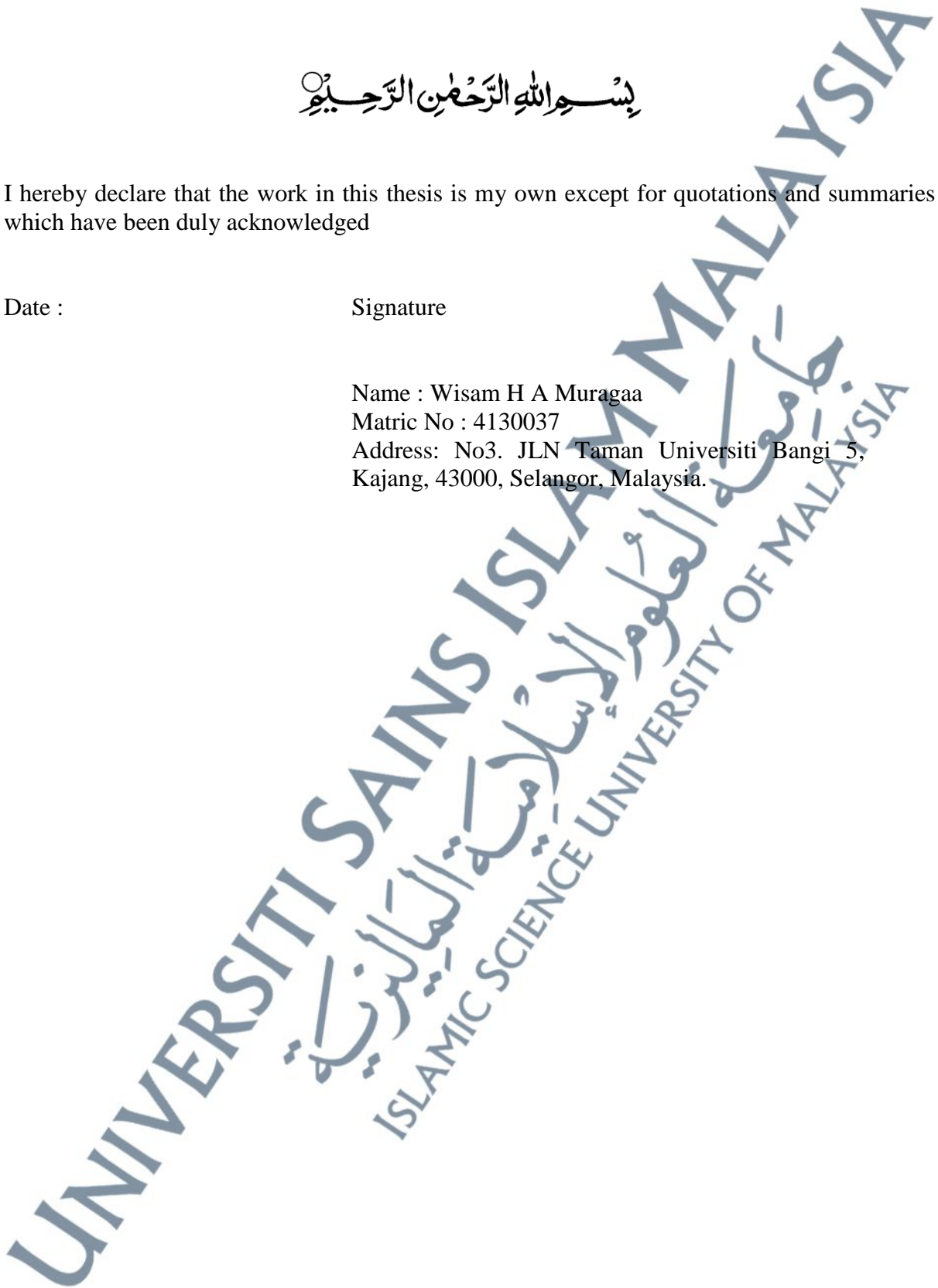
Date :

Signature

Name : Wisam H A Muragaa

Matric No : 4130037

Address: No3. JLN Taman Universiti Bangi 5,
Kajang, 43000, Selangor, Malaysia.



ACKNOWLEDGEMENTS

First and foremost, I would like to thank the “**Almighty ALLAH**” for giving me the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily. Without his blessings, this achievement would not have been possible.

I have great pleasure in acknowledging my gratitude to my supervisor **Prof. Dr. Kamruzzaman Seman**. He has been there providing his guidance and support at all times and has given me invaluable suggestions in my quest for knowledge. He has given me all the freedom to pursue my research, while silently and non-obtrusively ensuring that I stay on course and do not deviate from the core of my research. Without his able guidance, this thesis would not have been possible and I shall eternally be grateful to him for his assistance. I would like to express my gratitude to **Dr. Mohd Fadzli Marhusin** the co-supervisor. I would also like to express my gratitude to **Prof. Madya Dr. Nurlida Binti Basir** for her guidance and support.

My acknowledgment would be incomplete without thanking the biggest source of my strength, my family. My parents, whose love and guidance are with me in whatever I pursue. My prime sources of ideas my loving and supportive wife and my three wonderful children who provide unending inspiration. All these important people have contributed significantly to helping me reach this important stage in my life.

ABSTRAK

Sepanjang dekad yang lalu, serangan Pengedaran Penafian Perkhidmatan (DDoS) menjadi salah satu isu keselamatan Internet utama dan senjata pilihan untuk penggodam, pemerias siber, dan pengganas siber. Serangan DDoS mengganggu atau merendahkan perkhidmatan rangkaian (dengan mengurangkan kekurangan jalur lebar rangkaian atau kapasiti pemprosesan penghala) atau sumber mangsa (dengan melumpuhkan lebar jalur cakera atau pangkalan data, deskriptor fail, buffer, soket, kitaran CPU, memori) dan menghentikan pengguna yang sah daripada mengakses perkhidmatan Internet tertentu. Serangan tersebut memberi kesan kepada sumber-sumber mangsa supaya tidak dapat memberi respons kepada perkhidmatan yang diminta oleh pengguna yang disahkan. Di tengah meningkatkan bilangan serangan DDoS dan keupayaan penyerang untuk membangunkan jenis serangan untuk menembusi kaedah perlindungan tradisional, Perangkaian yang ditakrifkan perisian (SDN) telah muncul sebagai persekitaran alternatif untuk meminimumkan kerosakan serangan ini. Membolehkan kawalan rangkaian untuk diprogram secara langsung dan infrastruktur yang mendasar untuk diguna untuk aplikasi dan perkhidmatan rangkaian menjadikan rangkaian lebih fleksibel dan tangkas. Rangkaian yang ditakrifkan perisian merupakan model rangkaian baru yang menarik perhatian ramai penyelidik dalam isu keselamatan rangkaian hari ini. Pengesanan serangan DDoS menjadi lebih mudah jika kita dapat memanfaatkan ciri-ciri khas SDN seperti pemantauan kawalan ke atas infrastruktur, menanggalkan satah kawalan dari pesawat data dan konsep lalu lintas berasaskan aliran. Dalam kajian ini, kami menggunakan pengawal SDN untuk mengesan dan mengurangkan serangan DDoS. Penyelesaian pengesanan DDoS berasaskan SDN yang telah dicadangkan adalah pelbagai tetapi mengalami kemerosotan prestasi terutamanya, penyelesaian berasaskan pembelajaran mesin dan penyelesaian berasaskan entropi. Penyelesaian pengesanan DDoS berdasarkan teknik pembelajaran Mesin dan teknik entropi di SDN menderita sama ada meningkatkan penggunaan CPU atau meningkatkan penggera palsu. Degradasi prestasi dalam penyelesaian yang sedia ada disebabkan oleh kerumitan teknik yang digunakan untuk mengesan serangan DDoS di SDN serta parameter yang digunakan oleh teknik ini untuk membezakan paket DDoS. Penyelesaian yang ada tidak mengangap masa berlalu antara paket serangan berturut-turut sebagai parameter utama dalam mengesan serangan DDoS di SDN sebagai contoh. Selain itu, penyelesaian yang sedia ada hanya memberi tumpuan kepada mengesan serangan DDoS yang banjir dan gagal mencadangkan penyelesaian pertahanan yang dapat mengesan serangan DDoS yang berubah dari volum tinggi ke jumlah yang rendah pada masa serangan itu. Juga, beberapa penyelesaian dianggap sebagai pengesanan serangan DDoS kadar rendah di SDN. Dalam kajian ini, kami mencadangkan skim berasaskan masa yang berlalu, skim yang berkesan dan cekap untuk mengesan dan mengurangkan serangan banjir dan serangan kadar rendah di SDN. Skim berasaskan masa yang berlalu dilaksanakan pada pengawal POX dan dinilai berdasarkan senario serangan yang berbeza. Keputusan eksperimen mengesahkan bahawa, dibandingkan dengan penyelesaian berasaskan pembelajaran mesin yang berbeza dan penyelesaian berasaskan entropi yang disebut dalam penyelidikan ini, skim berasaskan masa yang berlaku mengurangkan overhead sehingga 50%, sambil memastikan 0% penggera palsu dan lebih dari 99.20% ketepatannya.

ABSTRACT

Over the last decade, Distributed Denial of Service (DDoS) attacks became one of the main Internet security issues and the weapon of choice for hackers, cyber extortionists, and cyber terrorists. A DDoS attack disrupts or degrades the network services (by depleting the network bandwidth or router processing capacity) or victim resources (by exhausting disk or database bandwidth, file descriptors, buffers, sockets, CPU cycles, memory) and stops the legitimate user from accessing a specific Internet service. Such attacks hog the victim's resources so that it cannot respond to the services requested by an authenticated user. Amid raising the number of DDoS attacks and the attackers' ability to develop attack types to penetrate traditional protection methods, Software-defined networks has emerged as an alternative environment to minimize the damage of this attack. Enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services make the network more flexible and agile. SDN is an emerging network model that attracted the attention of many researchers in today's networks security issues. The detection of DDoS attacks becomes much easier if we are able to take advantage of the SDN distinct characteristics such as the centralization of control over the infrastructure, decoupling of the control plane from the data plane and the flow-based traffic concept. In this research, we utilize the SDN controller to detect and mitigate DDoS attacks. The SDN-based DDoS detection solutions that have been proposed are various but suffer from performance degradation particularly, machine learning-based solutions and entropy-based solutions. DDoS detection solutions based on Machine learning techniques and entropy techniques in SDN suffer from either increase the CPU usage or increase false alarms. The performance degradation in the existing solutions is caused by the complexity of the techniques used to detect the DDoS attacks in SDN as well as the parameters used by these techniques to distinguish DDoS packets. The existing solutions do not consider the elapsed time between the successive attack packets as a key parameter in detecting DDoS attacks in SDN as an example. Furthermore, the existing solutions are only focused on detecting the flooding DDoS attacks and failed to propose a defense solution can detect the DDoS attacks that change from the high volume to the low volume at the time of the attack. Also, few solutions considered low-rate DDoS attacks detection in SDN. In this research, we propose an elapsed-time based scheme, an effective and efficient scheme to detect and mitigate flooding attacks and low-rate attacks in SDN. Elapsed-time based scheme is implemented on POX controller and evaluated under different attack scenarios. The experimental results confirm that, compared to different machine learning-based solution and entropy-based solutions mentioned in this research, elapsed-time based scheme reduces the overhead up to 50%, while ensuring 0% of false alarms and to more than 99.20% of the accuracy.

ملخص

خلال العقد الماضي ، أصبحت هجمات "رفض الخدمة الموزعة" (DDoS) واحدة من أهم مشكلات أمان الإنترنت والسلاح المفضل للمتسللين وابتزاز الإنترنت والإرهابيين عبر الإنترنت. يؤدي هجوم DDoS إلى تعطيل خدمات الشبكة أو تدهورها (عن طريق استنفاد عرض النطاق الترددي للشبكة أو سعة معالجة جهاز التوجيه) أو موارد الضحية (من خلال استنفاد عرض النطاق الترددي للقرص أو قاعدة البيانات أو واصفات الملفات أو المخازن المؤقتة أو المقابس أو دورات CPU أو الذاكرة) ويمنع المستخدم الشرعي من الوصول خدمة إنترنت محددة. تؤدي هذه الهجمات إلى خنق موارد الضحية بحيث لا يمكنها الاستجابة للخدمات التي يطلبها مستخدم معتمد. وسط زيادة عدد هجمات DDoS وقدرة المهاجمين على تطوير أنواع الهجمات لاخترق أساليب الحماية التقليدية ، برزت SDN كبنية بديلة لتقليل الضرر الناجم عن هذا الهجوم. تمكين التحكم في الشبكة لتصبح قابلة للبرمجة مباشرة والبنية التحتية الأساسية ليتم استخراجها للتطبيقات وخدمات الشبكة جعل الشبكة أكثر مرونة وخفة الحركة. الشبكات المعرفة بالبرمجيات هي نموذج شبكة ناشئ جذب انتباه العديد من الباحثين في قضايا أمن الشبكات الحالية. يصبح اكتشاف هجمات DDoS أسهل بكثير إذا تمكنا من الاستفادة من الخصائص المميزة لشبكة SDN مثل مركزية التحكم في البنية التحتية وفصل مستوى التحكم عن مستوى البيانات ومفهوم حركة المرور المستند إلى التدفق. في هذا البحث ، نستخدم وحدة تحكم SDN للكشف عن هجمات DDoS وتخفيفها. تعد حلول الكشف عن DDoS المستندة إلى SDN مختلفة ، ولكنها تعاني من تدهور الأداء خاصة ، والحلول المستندة إلى تعلم الآلة والحلول المستندة إلى الإنترنت. تعاني حلول اكتشاف DDoS المستندة إلى تقنيات التعلم الآلي وتقنيات الإنترنت في SDN من زيادة استخدام وحدة المعالجة المركزية أو زيادة الإنذارات الخاطئة. سبب تدهور الأداء في الحلول الحالية هو تعقيد التقنيات المستخدمة للكشف عن هجمات DDoS في SDN وكذلك المعلومات المستخدمة من قبل هذه التقنيات لتمييز حزم DDoS. لا تعتبر الحلول الحالية الوقت المنقضي بين حزم الهجوم المتتالية كمعلمة رئيسية في اكتشاف هجمات DDoS في SDN كمثل. علاوة على ذلك ، تركز الحلول الحالية فقط على اكتشاف هجمات DDoS الفيضانات وفشلت في اقتراح حل دفاعي يمكنها اكتشاف هجمات DDoS التي تتغير من الحجم الكبير إلى الحجم المنخفض في وقت الهجوم. أيضاً ، هناك حلول قليلة تعتبر اكتشاف هجمات DDoS ذات المعدل المنخفض في SDN. في هذا البحث ، نقترح خطة قائمة على الوقت المنقضي ، ونظام فعال وفعال لاكتشاف وتخفيف هجمات الفيضانات والهجمات ذات المعدل المنخفض في SDN. يتم تطبيق مخطط الوقت المنقضي على وحدة تحكم POX وتقييمه وفقاً لسيناريوهات الهجوم المختلفة. تؤكد النتائج التجريبية أنه بالمقارنة مع الحلول المستندة إلى تعلم الآلة المختلفة والحلول المستندة إلى الإنترنت المذكورة في هذا البحث ، فإن المخطط الزمني المستغرق يقلل من النفقات العامة بنسبة تصل إلى 50 % ، مع ضمان 0 % من الإنذارات الكاذبة وأكثر من 99.20 % من الدقة.

TABLE OF CONTENTS

AUTHOR DECLARATION	i
ACKNOWLEDGEMENT	ii
ABSTRAK	iii
ABSTRACT	iv
AL-MULAKHAS	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER 1 INTRODUCTION	1
1.1 Research Background	1
1.2 Problem Statement	2
1.3 Research Objectives	6
1.4 Significance of Research	6
1.5 Research Scope	9
1.6 Organization of the Thesis	11
CHAPTER 2 LITERATURE REVIEW	13
2.1 Introduction	13
2.2 Distributed Denial of Service (DDoS) Attacks	13
2.2.1 Operation of DDoS Attacks	14
2.2.2 Types Of DDoS Attacks	15
2.3 Software Defined Networking (SDN)	18
2.4 Openflow Protocol	21
2.4.1 Openflow Table Entry	21
2.4.2 OpenFlow Message Types	22
2.4.3 OpenFlow Operation	23
2.5 SDN Controllers	24
2.6 SDN Benefits	25
2.7 DDoS Attacks in SDN	26
2.7.1 Flow Table Overflow	29
2.7.2 Bandwidth Consumption	29
2.7.3 Controller Resource Saturation	30

2.8	Flooding DDoS Attack Detection Techniques in SDN	30
2.8.1	Machine Learning Techniques	30
2.8.2	Entropy Techniques	34
2.9	Low-Rate DDoS Attack Detection in SDN	39
2.10	Popular SDN-Based DDoS Attack Mitigation Methods	46
2.11	Related Work	48
2.12	Summary	49
CHAPTER 3 RESEARCH METHODOLOGY		51
3.1	Introduction	51
3.2	Analysis and Requirements	50
3.3	Designing The Elapsed-Time Based Scheme	53
3.2.1	Selecting a Validation Technique for a DDoS Research	53
3.2.1.1	Mathematical Models	54
3.2.1.2	Simulation	54
3.2.1.3	Emulation	55
3.2.1.4	Real Systems	55
3.2.2	Selecting a Network Simulator and Emulator	56
3.3.2.1	VirtualBox Software	57
3.3.2.2	Mininet Emulator	58
3.2.3	Selecting SDN Controller	59
3.2.3.1	POX Controller	60
3.2.4	Selecting a Programming Language	61
3.2.4.1	Python Programming Language	61
3.2.5	Selecting Parameters for Detecting and Mitigating DDoS in SDN	62
3.2.5.1	Parameters Used for DDoS Detection	63
3.2.5.2	Parameters Used for DDoS Mitigation	62
3.2.6	Simulation topology	66
3.3	Implementation of the Scheme Design	68
3.3.1	Experimental Hardware and Software	68
3.4	Testing and Evaluating the Implemented Scheme	68
3.4.1	Verification of The Network Emulator	69
3.4.2	Validation of the Network Emulator	70
3.4.3	Performance Metrics	71
3.4.3.1	Overhead (CPU usage)	68

3.4.3.2	Accuracy	71
3.5	Summary	73
CHAPTER 4 AN ELAPSED-TIME BASED SCHEM		75
4.1	Introduction	75
4.2	Phases of the Proposed Scheme	75
4.2.1	Phase One (Statistics Collection)	76
4.2.2	Phase Two (Attack Detection)	79
4.2.3	Phase Three (Attack Mitigation)	84
4.3	Summary	88
CHAPTER 5 RESULTS AND DISCUSSIONS		89
5.1	Introduction	89
5.2	Simulation Experiments	90
5.2.1	Experiment One (UDP Flood Scenario)	90
5.2.1.1	Test Case One (Normal Traffic Generation)	91
5.2.1.2	Test Case Two (Attack Traffic Generation)	93
5.2.1.2.1	Low Packets Scale	93
5.2.1.2.2	Large Packets Scale	97
5.2.1.3	Test Case Three (Mixture of UDP Flood Attack and Normal Traffic)	106
5.2.1.4	Scheme Performance Evaluation (UDP Flood Scenario)	116
5.2.1.4.1	Overhead (CPU Usage)	117
5.2.1.4.2	Accuracy	118
5.2.2	Experiment Two (Low-Rate SYN Scenario)	122
5.2.2.1	Test Case One (Normal Traffic Generation)	123
5.2.2.2	Test Case Two (Attack Traffic Generation)	125
5.2.2.2.1	LowRate Packets In The Short Term	125
5.2.2.2.2	LowRate Packets In The Long Term	130
5.2.2.3	Test Case Three (Mixture of low-rate Attack and Normal Traffic in long term)	137
5.2.2.4	Scheme Performance Evaluation (Low-Rate SYN Attack Scenario)	144
5.2.2.4.1	Overhead (CPU Usage)	144
5.2.2.4.2	Accuracy	145

5.2.3	Experiment Three (UDP Flood, Low-rate SYN and Normal traffic Scenario – long term)	146
5.2.3.1	Scheme Performance Evaluation (UDP flood, Low-rate SYN and Normal traffic scenario - long term)	164
5.2.3.1.1	Overhead (CPU Usage)	164
5.2.3.1.2	Accuracy	165
5.2.4	Experiment Four (UDP Flood, Low-rate SYN and Normal traffic Scenario – short term)	170
5.2.4.1	Scheme Performance Evaluation (Low-Rate SYN Attack Scenario)	184
5.2.4.1.1	Overhead (CPU Usage)	184
5.2.4.1.2	Accuracy	185
5.3	Discussion on The Results	186
5.4	Summary	189
CHAPTER 6 CONCLUSION		192
6.1	Study Summary	192
6.2	Study Contributions	194
6.3	Future Work	196
REFERENCES		197
APPENDIX A		204

LIST OF TABLES

Tables	Page
Table 2.1: Classification of DDoS attacks	17
Table 2.2: Controller Characteristics	25
Table 2.3: SDN Features in Defending DDoS Attacks	26
Table 2.4: Comparison Among Various Solutions of Machine Learning and entropy	41
Table 2.5: Various DDoS Mitigation Methods	46
Table 4.1: The Algorithm of the Elapsed-Time Based Scheme for Detecting and Mitigating DDoS Attacks in SDN	87
Table 5.1: Comparison of Results of the Elapsed-time Based Scheme to various Machine Learning Solutions	120
Table 5.2: Comparison of Results of the Elapsed-time Based Scheme to Various Entropy Solutions	121
Table 5.3: Comparison of Results of the Elapsed-time Based Scheme to Various Machine Learning Solutions	167
Table 5.4: Comparison of Results of the Elapsed-time Based Scheme to Various Entropy Solutions	169
Table 5.4: Comparison of results of the elapsed-time based scheme obtained in experiment three and four	189

LIST OF FIGURES

Figures	Page
1.1: Research Scope	10
2.1: DDoS Attack Structure	15
2.2: SDN Architecture	20
2.3: Flow Table Entry	22
2.4: The flow processing procedure in OpenFlow	24
2.5: DDoS Attack Scenario in SDN	28
2.6: DDoS Defender Flowchart	48
3.1: Research Methodology	52
3.2: Validation Techniques	53
3.3: Simulation Topology	67
3.4: Creating Simulation Topology Using Mininet CLI	69
3.5: Results of Using Testing Tools	70
4.1: Scheme Process	76
4.2: Packet Processing in OpenFlow Switch	77
4.3: Attack Detection	83
4.4: Attack Mitigation	85
4.5: An Elapsed-time based Scheme	86
5.1: The Performance Under Normal UDP Traffic	91
5.2: Summary of Normal UDP Traffic	92
5.3: Results Under Normal UDP Traffic	93
5.4.1: The First Results of the Low Scale of Packets	94

5.4.2: Results After Seventy Seconds	95
5.4.3: The Last Results	95
5.4.4: The Performance Under the Low Scale of Packets	96
5.4.5: Summary of the Low Scale of Packets	97
5.5.1: The First Results of the Large Scale of Packets	99
5.5.2: Results After Five Minutes	100
5.5.3: Results After Fifteen Minutes	101
5.5.4: Results After Twenty Five Minutes	103
5.5.5: Results after Thirty Minutes	104
5.5.6: The Performance Under the Large Scale of Packets	105
5.5.7: Summary of the Large Scale of Packets	106
5.6: POX Terminal Within the First Sixty Seconds	107
5.7: Results after Three Minutes	108
5.8: Results after Five Minutes	109
5.9: Results after Ten Minutes	110
5.10: Results after Twelve Minutes	111
5.11: Results after Fifteen Minutes	112
5.12: Results after Twenty Minutes	113
5.13: The Performance Under the Mixture of Traffic	114
5.14: The list of source IP addresses	115
5.15: The list of Destination IP addresses	115
5.16: Summary of the Mixture Traffic Generation	116
5.17: CPU Usage Under UDP Flood Scenario	117
5.18: An Elapsed-Time Based Scheme vs. Solutions Based on Different Techniques (machine learning technique, entropy technique)	119

5.19: The Performance Under Normal TCP Traffic	123
5.20: Summary of Normal TCP Traffic	124
5.21: Results Under Normal TCP Traffic	124
5.22: List of Source and Destination IP Addresses	125
5.23.1: Detecting Low-Rate Attack Packets from the First Second	126
5.23.2: Results After Three Minutes	127
5.23.3: Results After Five Minutes	127
5.23.4: Performance Under the Low-Rate Attack in the Short Term	128
5.23.5: Summary of Low-Rate Attack in the Short Term	129
5.23.6: List of the Source IP Addresses	130
5.23.7: List of the Destination IP Addresses	130
5.24.1: Detecting low-rate attack packets from the first second	131
5.24.2: Results After Five Minutes	132
5.24.3: Results after Fifteen Minutes	132
5.24.4: Results after Twenty Five Minutes	133
5.24.5: The Last Results	133
5.24.6: List of the Source IP Addresses	134
5.24.7: List of the Destination IP Addresses	134
5.24.8: Performance Under the Low-Rate Attack in the Long Term	136
5.24.9: Summary of Low-Rate Attack in the Long Term	136
5.25: POX Controller Terminal Within the First Twenty Seconds	137
5.26: Results after Twenty Seconds	138
5.27: Results after Ten Minutes	138
5.28: Results after Fifteen Minutes	139
5.29: Results after Twenty Minutes	139

5.30: The Last Results	140
5.31: List of the Source IP Addresses	141
5.32: List of the Destination IP Addresses	141
5.33: The Performance Under the Mixture of Traffic in the Long Term	143
5.34: Summary of the Mixture Traffic in the Long Term	143
5.35: CPU Usage Under Low-rate Scenario	145
5.36: POX Controller Terminal Within the First Eighty Seconds	147
5.37: Results after Eighty-Two Seconds	148
5.38: Results after Three Minutes	149
5.39: Results after Five Minutes	150
5.40: Results after Seven Minutes	151
5.41: Results after Ten Minutes	152
5.42: Results after Thirteen Minutes	153
5.43: Results after Fifteen Minutes	154
5.44: Results after Eighteen Minutes	155
5.45: Results after Nineteen Minutes	156
5.46: The Last Results	157
5.47: List of the Source IP Addresses	160
5.48: The Performance Under the Mixture of UDP Flood, Low-Rate SYN, and Normal Traffic	163
5.49: Summary of the Mixture Traffic (long-term)	164
5.50: CPU Usage Under the Mixture Traffic	165
5.51: An Elapsed-Time Based Scheme vs. Solutions Based on Different Techniques (machine learning technique, entropy technique)	166
5.52: POX Controller Terminal Within the First Hundred Seconds	171

5.53: Results after Hundred-five Seconds	172
5.54: Results after Two Minutes	173
5.55: Results after Three Minutes	174
5.56: Results after five Minutes	175
5.57: Results after Seven Minutes	176
5.58: Results after Eight Minutes	177
5.59: The Last Results	178
5.60: List of the Source IP Addresses	181
5.61: The Performance Under the Mixture Traffic (short term)	183
5.62: Summary of the Mixture Traffic (short-term)	184
5.63: CPU Usage Under the Mixture Traffic	185