Reinforcement Learning for Resilient DDoS Defence

Jeremy Michael Pattison

637841

Supervisors:
Prof. Christopher Leckie
Dr. Sarah Monazam Erfani

A thesis presented for the subject of
Computer Science Research Project COMP90070
Conventional research project of 75 credit points

Department of Computing and Information Systems
The University of Melbourne
June 2019

Abstract

Distributed Denial of Service (DDoS) attacks seek to restrict the availability of resources at the server to disrupt legitimate use of service. Sophisticated attacks continue to bypass or exploit current DDoS defences. Standard evaluation practices simulate a proposed DDoS defender against expected network traffic. The approach struggles to replicate realistic network behaviour and does not consider future developments of user behaviour or the existence of the intelligent attacker. Rate-limiting is a DDoS defence approach used to manage congestion attacks when the attacker floods the network through volume of traffic. This thesis investigates the role of Reinforcement Learning to aid the formulation of effective rate-limiting defenders.

The thesis explores the potential of a rate-limiting defender to adapt to its deployed network. We draw on the Reinforcement Learning MARL rate-limiter to establish MARL's suitability as a replacement for the comparative Fair Throttle which uses a Control Mechanism by contrast. Through examining the effectiveness of MARL we challenge prior comparisons where the Fair Throttle is demonstrated to outperform MARL against variable traffic.

The second problem addressed is the establishment of realistic evaluation, current automated approaches risk unforeseen vulnerabilities by neglecting the intelligent attacker. We propose the use of Reinforcement Learning to model an intelligent DDoS attacker. Our Intelligent DDoS Adversary (IDA) can be used as a cost effective evaluation tool to determine how a rate limiting DDoS defender will perform against an intelligent attack. IDA is demonstrated to outperform standard attacks against all rate-limiters in all topologies when provided the same attacking resources. IDA is used to extend the analysis for MARL and the Fair Throttle where IDA identifies a new vulnerability in MARL that was not apparent through standard practice methodology. By demonstrating that Reinforcment Learning can be used to evaluate new rate-limiting defences we conclude by establishing a series of challenges MARL must satisfy before being considered a viable rate-limiter.

Declaration

I certify that

1. *This thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.*

2. *Where necessary I have received clearance for this research from the University's Ethics Committee and have submitted all required data to the Department.*

3. *The thesis is 16,218 words in length (excluding text in images, table, bibliographies and appendices).*

Jeremy Michael Pattison
Computing and Information Systems
University of Melbourne, Australia
June 11, 2019

Contents

Declaration **1**

1 Introduction **8**
1.1 Focus 9
1.2 Thesis Overview 11

2 Background and Literature Review **14**
2.1 Denial of Service Attacks 14
2.2 Reinforcement Learning 16
2.2.1 Markov Decision Process 16
2.2.2 Temporal Difference Approaches 17
2.2.3 Function Approximation 18
2.2.4 Multi-Agent Learning 19
2.3 Reinforcement Learning For Defence 20
2.3.1 Establishment of Appropriate Goals 20
2.3.2 Reinforcement in Intruder Detection 21
2.3.3 Reinforcement in Congestion Control 23
2.4 Resilience to Atypical Attacks 26
2.4.1 Realistic Evaluation 27
2.4.2 Modelling an Intelligent Adversary 27
2.4.3 Evaluation Through Intelligent Attacks 27
2.5 Summary 28

3 Resistant Defenders **31**
3.1 Problem Statement 32
3.1.1 Network Model 33
3.2 Investigated Defenders 34

3.2.1 MARL . . . 34
3.2.2 MARL Adaptations . . . 37
3.2.3 Comparative Baseline . . . 38
3.3 Simulation Results . . . 39
3.3.1 Evaluation Criteria . . . 40
3.3.2 3 Agent Topology . . . 41
3.3.3 6 Agent Topology . . . 43
3.3.4 Removing the Bottleneck . . . 45
3.4 Discussion and Conclusion . . . 46
3.4.1 Analysing MARL . . . 46
3.4.2 Conclusion . . . 47

4 Modelling an Intelligent DDoS Adversary 49
4.1 Problem Statement . . . 50
4.1.1 Extension to Network Model . . . 50
4.2 Intelligent DDoS Adversary (IDA) . . . 51
4.2.1 Structure . . . 53
4.2.2 Action Space . . . 53
4.2.3 State Space . . . 53
4.2.4 Reward Structure . . . 54
4.3 Experimental Design . . . 54
4.3.1 Training . . . 55
4.4 Simulation Results . . . 55
4.4.1 Evaluation Criteria . . . 56
4.4.2 Evaluation results . . . 57
4.4.3 Analysing IDA . . . 61
4.4.4 Isolating Defence Mechanisms . . . 64
4.4.5 Increasing Handlers . . . 65
4.5 Discussion and Conclusion . . . 66

5 Conclusion and Future Work 68
5.1 Overview . . . 68
5.2 Future Work . . . 71
5.2.1 Designing Adaptive Rate-Limiters . . . 71
5.2.2 Adapting for a Continuous Action Space . . . 71
5.2.3 Addressing Scalability Concerns . . . 71
5.2.4 Application for a General Network Evaluator . . . 72

5.2.5 Incorporation into Standardised Test Beds 72
5.2.6 Threat of an Online Adversary 73

List of Figures

1.1 A generalisation of the design process of the attacker and defender. 10

2.1 General DDoS Structure [Lin and Tseng, 2004] 15
2.2 Representation of a Reinforcement Learning Agent [Sutton and Barto, 2018]. 17

3.1 Development cycle for DDoS defence design. 32
3.2 The 6 Defender Topology showing the roles of all agents. 33
3.3 Depiction of the learning work flow of MARL. 35
3.4 Comparison of prior work measuring the mean percentage of traffic served in three agent topology. 42
3.5 The legitimate traffic served is measured in a six agent topology. 44
3.6 The versatility of the defenders is examined by removing the server's capacity to mitigate congestion. 45

4.1 Generalised illustration of the adversary's interaction with the network defender. 51
4.2 Detailed illustration of the interactions of all actors in the learning of optimal attack patterns. 52
4.3 Percentage of traffic served during training of IDA against the Fair Throttle. 56
4.4 Comparison of legitimate packets served by both IDA and standard attacks. 59
4.5 Divergence of network disruption between IDA and Constant Attack. 62
4.6 Observed attack patterns by IDA against the Fair Throttle and MARL 63

4.7 IDA evaluates the capacity of the defenders to avoid congestion on the 9 agent topology 64

List of Tables

2.1 A summary of Reinforcement Learning's contributions to DDoS Defence 30

3.1 Comparison of legitimate packets served by the Fair Throttle and Fixed Throttle. 39
3.2 Summary of different defenders evaluated. 41
3.3 Evaluated Attacks for Chapter 3. 41

4.1 Summary of different defenders to be evaluated by IDA. 58
4.2 Evaluated Attacks for Chapter 4 58
4.3 Percentage of legitimate traffic served as the number of Handlers is increased. 65

Chapter 1

Introduction

A Denial of Service (DoS) attack represents "an explicit attempt by hackers to prevent legitimate use of a service" often by consuming available resources [Mirkovic and Reiher, 2004]. DoS attacks overwhelm the victim, forcing the network to discard incoming traffic or risk system failure. A potent extension of the DoS threat, known as a Distributed Denial of Service (DDoS) attack, involves multiple attacking agents coordinating against a victim amplifying the potential bandwidth delivered [Peng et al., 2007]. DDoS attacks pose significant risks to businesses and infrastructure; it has been estimated that a successful DDoS attack incurs a cost of $500,000 on the targeted business [Matthews, 2014]. The severity of the DDoS threat continues to grow requiring further sophistication in network defence. In 2018 there was an observed DDoS attack emitting traffic rates of up to 1.7 Tbps which marked a 273% of the maximum rate observed in 2017 [NetScout, 2018].

The fundamental challenge with DDoS defence is designing defence solutions resistant to an evolving landscape. Defence systems are often effective against the attacks they were designed for but struggle against developing network behaviour [Shiva et al., 2010]. Despite numerous proposed defensive mechanisms we continue to observe attackers evolve their approach to bypass current defences. For example, the Cloudpiercer tool enabled researches to bypass cloud based network defences for 71.5% of tested websites [Vissers et al., 2015]. An ill-designed defender can be exploited by an attacker, a survey found 43% of companies in 2018 reported that their defence mechanisms contributed to their DDoS outage [NetScout, 2018].

Network defence is commonly designed and evaluated in offline emulation

before being deployed into a live environment to minimise the risk of a successful attack. If the emulation of the network is not realistic we risk unseen vulnerabilities not apparent until the defender is deployed. Modelling an attack can be difficult due to the large variation in potential attacks; e.g. attacks can differ by geographical distribution, rate of each attacking agents and the design of individual DDoS packets [Peng et al., 2007]. There exists significant literature understanding the dynamics of DDoS attacks [Wang et al., 2017, Mirkovic and Reiher, 2004, Peng et al., 2007] which in turn can simulate as to how an attacker would likely operate [Kotenko and Ulanov, 2014]. Designs are reactive to observed attacks and designed to accommodate network behaviour of the present therefore struggle to adapt to future developments in network behaviour. The two research goals we contribute to are:

(i) Providing the ability for a defender to adapt to its network environment.

(ii) The establishment of realistic evaluation that considers the intelligent attacker.

Reinforcement Learning has been proposed as a means to combat new and modified threats due to its ability to continue learning while active [Cannady, 1998]. Reinforcement Learning is a paradigm where an agent learns an optimal policy through interaction with the environment and can formulate a policy despite limited information and delayed rewards. In the context of network defence it can provide an adaptive defence mechanism that continues to update its policy after deployment. This thesis investigates the challenge of accounting for developments in network behaviour though two differing lenses. First it seeks to extend a previously proposed Reinforcement Learning rate-limiter, assessing the viability of the approach for an adaptive network. Then shifts the focus to the attacker and investigate the utility of Reinforcement Learning to extend evaluation practices through simulating the intelligent adversary.

1.1 Focus

Figure 1.1 provides a generalised overview of the dynamic between the defender and the attacker. Both actors observe the opposing actions and update their behaviour based on their belief how the opposing party will react. This behaviour can be modelled into a Reinforcement Learning problem. Automating the development of defence policy would allow defenders to adapt quickly to

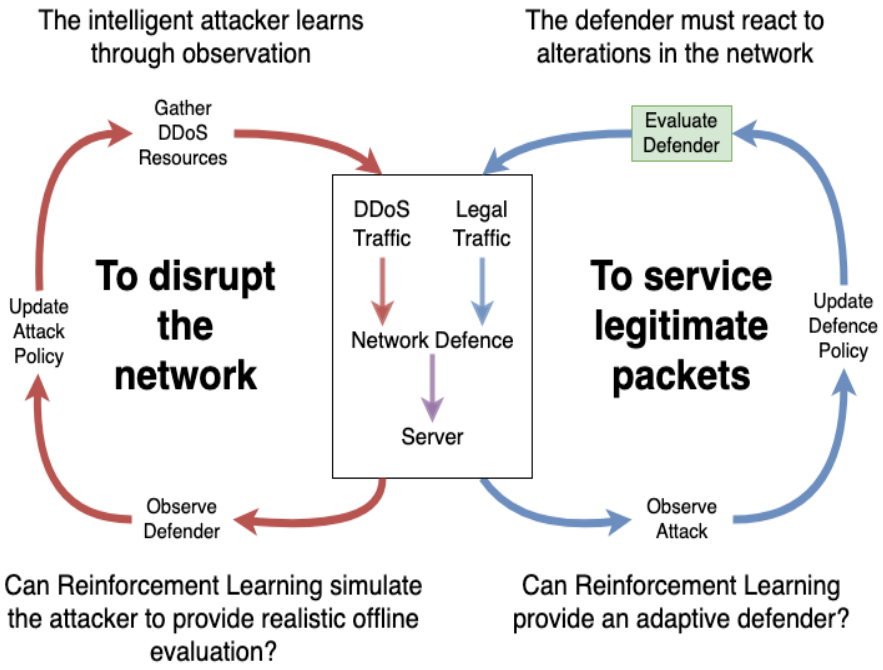


Figure 1.1: A generalisation of the design process of the attacker and defender. By simulating this process through Reinforcement Learning we enable the adaptive defender and realistic evaluation

developing threats. By simulating the design of intelligent attacks we can extend the evaluation of proposed defences against atypical attack patterns that exceed databases of observed attacks.

Rate-limiting is an Attack Reaction approach that seeks to reduce congestion whilst minimising disruption of legitimate traffic. This thesis investigates the effectiveness of the MARL Reinforcement Learning rate-limiter [Malialis and Kudenko, 2013b] and the Fair Throttle which uses a Control Algorithm [Yau et al., 2005]. MARL has been shown to either match or outperform the Fair Throttle in simulations that involved variable traffic [Malialis and Kudenko, 2013b]. By using Reinforcement Learning we allow a system that learns to update its policies over an evolving landscape. The system learns a throttling policy for the network's environment allowing an immediate response rather than having to converge to an appropriate throttle. Further investigation is required to establish the effectiveness and versatility of MARL. We seek to investigate

whether the MARL agent can be a viable alternative to the comparative Fair Throttle, and if not, determine how MARL can be an feasible rate-limiter for network defence.

Q1: Can MARL be made a suitable replacement to the Fair Throttle?

The second challenge we aim to address is providing insight on how an adversary may choose to attack the system. It is standard practice to evaluate a new DDoS proposal against observed attacks. The approach neglects to account for an intelligent attacker that may seek to exploit an existing mechanism. Therefore we risk unforeseen vulnerabilities that may fail to impede the attack or worse contribute to the outage. Individual analysis of proposed mechanisms, either through penetration testing or game theoretical approaches, can alert the researcher of an undiagnosed vulnerability however this approach is expensive and does not provide comparative evaluation of alternative approaches. Fig 1.1 illustrates that the design process of the attacker is similar to the defender thus could be replicated through Reinforcement Learning. In this thesis we propose using Reinforcement Learning to simulate the learning process an intelligent adversary to predict future attacks and contribute to realistic evaluation. By extending the comparison between MARL and the Fair Throttle we seek to establish the utility of the approach as an adaptive evaluator.

Q2: Can Reinforcement Learning be used to develop automated tools for testing the effectiveness of existing defences?

The focus of previous Reinforcement Learning research has been on creating an ideal or adaptive defender. In Chapter 3 this is expanded through investigation of MARL with the introduction of Deep Reinforcement Learning. Chapter 4 delves into the design of an optimal attacker, where Reinforcement Learning is used to simulate an intelligent attacker to evaluate existing defences.

1.2 Thesis Overview

Chapter 2

This chapter provides a basis of DDoS attacks and the capacity to improve current defence design. It seeks to outline the challenges of DDoS threats presented in existing literature, as well as introduce the theoretical background of Reinforcement Learning. Drawing on the MARL system as a case study,

this thesis highlights a potential vulnerability and identifies two proposals to increase the performance of the system.

Chapter 3

Chapter 3 provides further analysis of the previously proposed MARL system. We extend the comparative analysis of the MARL and Fair Throttle rate-limiters in a series of topologies and environments where we challenge assumptions of the ideal environment. Our analysis addresses *Q1* by including adaptations of MARL where we investigate under what conditions does MARL outperform the Fair Throttle.

- We introduce two adaptations of MARL that improve performance against constant traffic.
- We investigate the utility of MARL and Fair Throttle in a series of topologies and environments.
- We challenge previous comparative evaluations by finding the Fair Throttle outperforms MARL against variable traffic rates.
- We establish a versatility challenge by identifying MARL is reliant on the existence of a bottleneck against variable traffic rates.

Chapter 4

Chapter 4 addresses *Q2* which is the challenge of modelling an intelligent attack for a DDoS evaluation. We provide a novel approach that extends considerations to include both the provided resources and the functionality of the defending agent. Our adversary functions as an evaluation tool and is compatible with standard practice. The adversary utilises Deep Reinforcement Learning to interpret an extensive state, it is architecturally similar to the general DDoS topology presented by *Lin et al.* [Lin and Tseng, 2004].

- We introduce a novel evaluation metric which simulates the behaviour of an intelligent attacker.
- We demonstrate one such model where we evaluate a series of rate-limiting defenders against an intelligent adversary.
- Our adversary contributes to current literature by identifying an unexplored vulnerability of the MARL defender which is not apparent through standard evaluation practices.

Chapter 5 We summarise the contributions of this thesis and elaborate on challenges faced by MARL and the intelligent adversary. We identify future areas of research including the potential to expand the rate-limiting adversary to an automated penetration testing tool.

Chapter 2

Background and Literature Review

2.1 Denial of Service Attacks

The goal of the DDoS attack is to prevent legitimate use of service [Mirkovic and Reiher, 2004]. DoS attacks have been studied throughout environments including operating systems [Gligor, 1984], wireless networks [Li et al., 2017] and the internet [Needham, 1994]. Attacks can be generally categorised into Protocol or Bandwidth attacks [Peng et al., 2007]. Protocol attacks often involve "carefully crafted packets" that exploit a vulnerability, each packet can be expected to request a greater amount of resources than a typical packet [Peng et al., 2007]. The 'SYN Flood', which exploits the TCP three-way handshake, would be considered such an attack as it restricts a server's future connections [Peng et al., 2007]. Protocol based attacks can often be mitigated by "modifying the misused protocols or deploying proxies" [Mirkovic and Reiher, 2004]. Bandwidth Attacks, also known as 'Brute Force' attacks seek to overload a network by sheer volume of seemingly legitimate packets [Mirkovic and Reiher, 2004]. Packets from an HTTP Flood designed to mimic user behaviour can be difficult to differentiate from user traffic presenting a challenge for network defence [Peng et al., 2007].

Our focus in this thesis centres on Brute Force DDoS attacks, unless otherwise stated, future references to the DDoS attack consider a Brute Force DDoS attack where there is a limited ability to distinguish legitimate packets from

attacking packets.

DDoS Structure

Typical DDoS attacks are comprised of an attacker controlling a large botnet, also known as zombies, through adversarial handlers [Peng et al., 2007]. The hierarchical design allows an attack to originate from a geographically diverse area hindering countermeasures such as traceback or blocking. Figure 2.1 shows the DDoS structure where the attacker has delegated the attacking agents to its Handlers. The increased size lessens the number of requests per agent needed to incapacitate the victim reducing traffic disparities between attackers and users. Large botnets are able to attack a server with each bot attacking in a manner reflective of typical users, such attacks are almost indistinguishable from flash crowds, an event where many legitimate users simultaneously access the server [Peng et al., 2007].

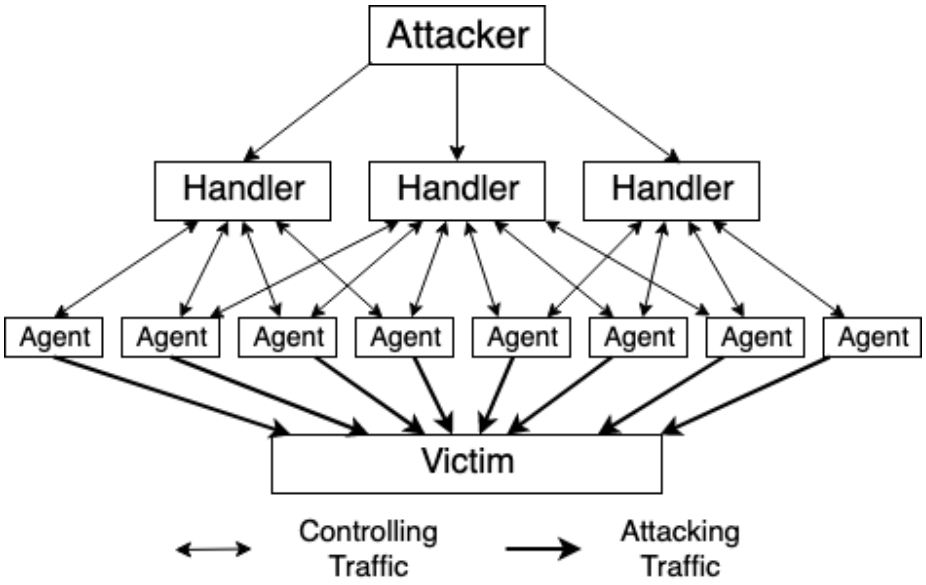


Figure 2.1: General DDoS Structure [Lin and Tseng, 2004]

Defence Mechanisms

Peng et al. categorised common defence mechanisms into 'Attack Prevention', 'Attack Detection', 'Attack Source Identification' and 'Attack Reaction' [Peng

et al., 2007]. Attack Prevention is the first layer of the defence aiming to stop attacking packets before damage is caused. Attack Detection is used to alert the system whether it is currently encountering a DoS attack allowing the system to initiate reactive measures. Whilst server degradation might be considered an easy metric to evaluate the presence of a DDoS attack, further analysis is required to differentiate Flash Crowd traffic leading to potential false positives [Peng et al., 2007]. Anomaly detection aims to identify "if the monitored traffic behaviour does not match the normal traffic profile that is built using training data" [Peng et al., 2007]. Identifying the source of the attack is a preventative measure to allow targeted reactive measures. Attack Reaction refers to the response by the network or server to mitigate an ongoing attack. Common approaches involve congestion control to reduce incoming bandwidth or re-routing adversarial traffic to a 'blackhole'.

2.2 Reinforcement Learning

Reinforcement Learning is a Machine Learning approach where an agent learns through interaction with the environment. After each action the environment is updated and the agent is provided a reward reflecting the utility of the new environment, the agent learns an optimal policy by seeking to maximise its total reward given an environment. The approach differs from supervised learning as the agent is rewarded through indirect rewards calculated from environment feedback as opposed to labelled examples.

2.2.1 Markov Decision Process

A Markov Decision Process (MDP) is a mathematical formalisation of Reinforcement Learning, modelling the relationship with the agent and environment [Sutton and Barto, 2018]. A state is a representation of the environment that is provided to the agent. For every potential state there is an associated set of agent actions, transition probabilities dictating the result of each action and an associated reward for the state, often represented as <State, Action, Transition Probabilities, Reward>. By way of example, for state S_1, action a_1 might have a 33% probability of leading to a state with an associated reward of 0.8 and a 67% chance of leading to a state with a reward of -1. The policy dictates what action the agent should perform given a particular state. The challenge is to find a policy that maximises the agent's reward over time. When the agent

is provided the entire MDP model, the task can be solved through Dynamic Programming, approximation approaches are required for large MDPs or where the agent is not provided the full model [Sutton and Barto, 2018].

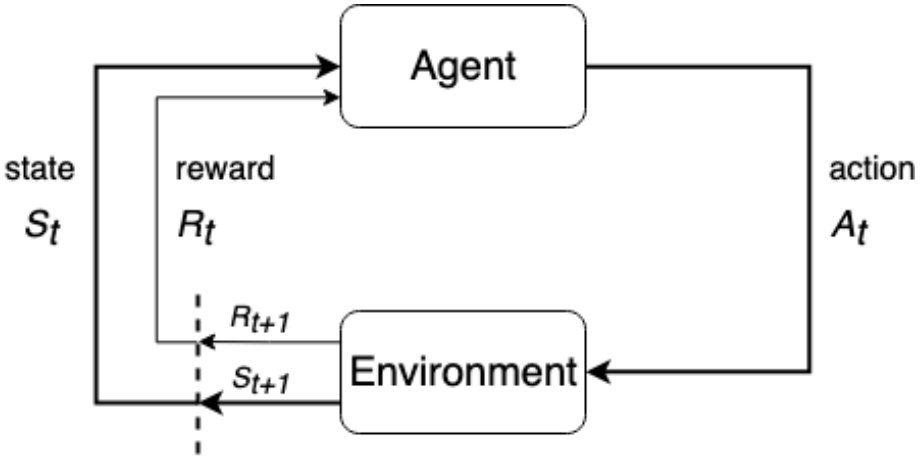


Figure 2.2: Representation of a Reinforcement Learning Agent [Sutton and Barto, 2018].

2.2.2 Temporal Difference Approaches

Temporal Difference Learning is a bootstrap approach that allows an agent to formulate a policy despite limited information whilst providing intermediate rewards [Sutton and Barto, 2018]. The agent learns through simulation with the environment, a graphical representation is provided in Figure 2.2, each step it chooses an action for the given state, the action is then modelled into the environment where the world is updated and it receives a reward. This approach allows the agent to learn the expected reward for each state-action combination. The transition probabilities are encapsulated into our policy when the agent experiences differing rewards for the same action-state combination. Thus the agent formulates a policy of reward maximisation for any given state.

The policy is formulated as a Q-table where we map each action for a given state with an expected reward. Once the policy has been learnt, each step the agent chooses the action with the expected highest reward, during training we must balance the competing objectives of exploring action-state pairs known to be advantageous and continuing to explore conceived non-profitable action-states as our expected reward may be incorrect, this is known as the

Exploration-Exploitation trade-off. Often we are interested in long term rewards as opposed to the immediate reward received, the use of Temporal Difference Learning allows the Q-table to incorporate the expected reward on the new state into the reward calculation. Let α be our learning rate and γ be our discount value which incorporates future rewards, after each action during training the Q-table is updated by the following formula:

$$Q(S_t, A_t) = V(S_t, A_t) + \alpha[R_{t+1} + \gamma Q_{next} - Q(S_t, A_t)]$$

Rewards from states several actions away are implicitly incorporated by Q_{next} which represents our expected reward for the next state. Q-learning and SARSA learning are two popular learning approaches that differ only by their estimation of Q_{next}. Q-learning assigns $Q_{next} = max_a Q(S_{t+1}, a)$, that is, it assumes that the following move would be the optimal move. An alternative approach is to consider the actual outcome as opposed to the optimal outcome, this is particularly important during exploration where we expect the agent to make sub-optimal moves. This is reflected in the SARSA update rule $Q_{next} = Q(S_{t+1}, A_{t+1})$, that is it uses the Q-value from action-state that the agent performs. During training, the Q-learning approach will often overestimate the utility of a new state, whilst SARSA conversely underestimates the value [Sutton and Barto, 2018]. The exploration-exploitation trade-off, where agents try to avoid believed disadvantageous states, leads to what is referred to as maximisation bias. Double Learning is a learning approach that avoids maximisation bias by learning two estimates as opposed to one for every state, it is compatible with Temporal Difference learning approaches such as Q-learning and SARSA [Sutton and Barto, 2018].

2.2.3 Function Approximation

Where there are a large number of states, or the states are continuous, it is necessary to approximate the Q-table. Linear Function Approximation and Deep Reinforcement Leaning are two popular approaches for approximating the state and the representation of the Q-table.

Tile-coding is a popular Linear Function Approximation method which represents the state with a series of overlapping tiles. At each time the agent exists in simultaneous states, the optimal action is the action that has the total highest expected reward averaged over all existing states.

A growing area of research is approximating the Q-table with a non-linear neural network. Deep Reinforcement Learning aims to combine the capacity of Deep Learning to interpret large states with Reinforcement Learning. The Deep Q-Network (DQN) algorithm is a Q-learning implementation of Deep Reinforcement Learning [Mnih et al., 2015]. Like traditional Q-learning, DQN suffers from maximisation bias, whereas a Double Learning DQN algorithm (DDQN) avoids maximisation bias and was demonstrated to outperform humans against a series of Atari video games [Van Hasselt et al., 2016].

A liability of Deep Reinforcement Learning is that small changes to the Q-value can significantly change the neural network leading to stability issues, this is heightened when exposed to correlated training examples [Mnih et al., 2015]. Experience Replay is one approach to provide stability that randomises updates by continuously sampling past experiencing throughout training.

2.2.4 Multi-Agent Learning

We now broaden our analysis to consider problems where there exist multiple agents sharing the same goal, this may result in each agent exposed to different states. Often a single learning agent, which interprets a global state and then instructs all agents, can be discounted due to scalability concerns and communication delays [Panait and Luke, 2005]. Agents explore simultaneously to formulate a policy to maximise their own reward. Despite the lack of communication, agents can formulate a collaborative policy by factoring in other agents as part of their environment, over time expected rewards would be dependent on other agent's behaviour.

Multi-Agent-Reinforcement-Learning can often suffer scalability concerns stemming from the reward structure. A global reward, where all agents are rewarded equally, can be used to encourage collaborative behaviour however this risks rewarding agents for inconsequential actions where another agent was responsible for the bulk of the work [Panait and Luke, 2005]. By contrast, rewarding each agent by their individual action can often lead to competing behaviour where agents will proceed with the action that maximises their personal reward regardless of the expected reward for the system [Panait and Luke, 2005].

The application of Deep Reinforcement Learning (DRL) presents a further complication. Recall both that DRL often uses past experience to provide stability, and that each agent models other agents as part of their environment.

As each agent continues to update their policies the existence of simultaneous learners represents a changing environment, repetitive sampling from past situations that are inconsistent with the present scenario directly trains our agent with inaccurate rewards leading to suboptimal outcomes [Foerster et al., 2017]. An empirical study, which involved two opposing teams of three DRL agents each, found despite a changing environment Experience Replay provided greater returns when compared to a scenario without experience replay. The authors also provided two mechanisms that discouraged sampling experiences from significantly different environments [Foerster et al., 2017].

Reinforcement Learning is a powerful tool that allows the training of an agent policy through indirect rewards. When combined with Deep Learning it has the capacity to interpret large and non-linear state-spaces. It should, however, not be considered a silver bullet to tackle a difficult task. The clarity of accurate rewards is a key consideration whether Reinforcement Learning is an appropriate approach to solve a challenge. Furthermore, the presence of multiple agents can obscure reward distribution which is exasperated for learning approaches that require experience replay.

2.3 Reinforcement Learning For Defence

The ability to interpret and learn from large sets of data has seen Machine Learning become an increasingly popular area of research for network defence. Reinforcement Learning's capacity to associate its interpretation of the state with prospective actions, as well as the capacity for continuous learning, has seen efforts to combine Reinforcement Learning with Intruder Detection and Attack Reaction. Although our focus is Congestion Control, an Attack Reaction approach, we also provide a brief review of previous research introducing Reinforcement Learning to Intruder Detection as well as Attack Reaction.

2.3.1 Establishment of Appropriate Goals

First, allow us to consider the requirements a Reinforcement Learning solution to internet security must satisfy. As the potential of Reinforcement Learning is not limited to a specific area of DDoS defence the overarching goals should be generalised. The establishment of generalised metrics has been investigated by *Mirkovic et al.*; the following is a brief set of goals that the authors deemed that a general DDoS system must satisfy [Mirkovic et al., 2009, Mirkovic et al.,

2006]:

- The primary goal is to demonstrate that the proposed mechanism is effective.
- "Memory and CPU costs must be quantified".
- The defence must be evaluated for scalability.
- The defence mechanism must be resilient to attempts by the adversary to be specifically targeted. Resilience considers the likelihood and impact of the system being gamed or failing.

Reinforcement Learning provides a mechanism to allow a system to learn effective policies to combat a threat. Often systems are designed and evaluated in unrealistic network environments [Mirkovic et al., 2009], the ability for the system to continue to learn after deployment allows the system to develop a policy catered for the deployed network. Reinforcement Learning adopts sub-optimal policies if the evaluated and trained environments differ [Sutton and Barto, 2018]. We consider online learning necessary for the defender as realistic emulation of networks is an ongoing research challenge [Mirkovic et al., 2009]; even if we were able to replicate the expected network behaviour, online learning would still be required as user behaviour diverges from the model overtime. To allow continuous learning the system must be provided an accurate reward calculator that continues to provide the system rewards based on the objectives. Thus, the Reinforcement Learning approach must consider the same goals for the reward calculator. This is necessary as a poorly designed reward can be subject to 'reward hacking' where the system performs a series of negative actions, such as blocking all traffic, to maximise an assigned reward, such as mitigating DDoS traffic [Hadfield-Menell et al., 2017]. Accordingly, we must consider both the effectiveness and versatility of the reward in a live environment.

2.3.2 Reinforcement in Intruder Detection

Recollect that Intruder Defence refers to identifying whether the victim is currently exposed to a DoS attack. The quickly changing nature of internet attacks leads to the potential for a victim to experience an attack to which it has had little to no prior exposure. Detection approaches that rely on expert solutions require frequent updates to resolve emerging vulnerabilities, many

Machine Learning approaches require periodic relearning as new data is added [Cannady, 1998]. An early use of Reinforcement Learning in the context of DoS attacks by *Cannady* allowed continued training of a DoS detector after deployment. By formulating a reward from the server's functionality, the neural network was able to adapt to previously unseen DoS attacks [Cannady, 1998]. A similar system, which used a decentralised Reinforcement Learning mechanism for Intruder Detection, was proposed by *Servin and Kudenko* where the agent was rewarded for correctly identifying DoS attacks [Servin and Kudenko, 2008]. Both systems are rewarded if they perform actions given the presence or absence of a DDoS attack.

These systems lack the establishment of an effective reward mechanism to allow online learning. The proposal by *Cannady* uses the presence of server congestion to calculate the reward which lacks a capacity to differentiate DDoS attacks from Flash Crowds, in this case it would be just as effective to measure congestion rather than train a system that learns traffic patterns that arise when congestion occurs. The reward mechanism is omitted in the proposal by *Servin and Kodenko*, thus the mechanism must learn its policy through network simulation. Further research is required to investigate how the application performs in a real environment that differs from the trained network simulation. Fundamentally, the generation of an accurate reward would require the solving the initial problem, that is determining whether the system is encountering a DDoS attack. As a result it would be more efficient to use the reward calculator to perform the task unless there existed a reliable calculator restricted to delayed rewards.

A more appropriate use of Reinforcement Learning in the context of Intruder Detection was to address scalability concerns for communication between a decentralised HMM system [Xu et al., 2007]. Here the Intruder Detection is performed by the HMM system whilst the Reinforcement Learning system is rewarded dependent on if the inter-agent communication was beneficial. The reward is deterministic in a network environment, the use of Reinforcement Learning decreased network overhead by restricting excessive communication and was therefore remains a prospective implementation of Reinforcement Learning; *Nguyen and Choi* argued that the HMM mechanism has had insufficient evaluation within a realistic network environment [Nguyen and Choi, 2010].

2.3.3 Reinforcement in Congestion Control

Rate Limiting is an Attack Reaction approach that seeks to drop upstream packets to manage server congestion whilst minimising disruption to legitimate traffic. A Multi-Agent-Reinforcement-Learning rate limiter, named MARL, was produced by *Malialis and Kudenko* [Malialis and Kudenko, 2013b] and is investigated in this thesis. To properly determine the application's effectiveness we first analyse popular rate-limiting systems that MARL seeks to replace.

ACC

Aggregate Congestion Control (ACC) seeks to generalise traffic into aggregates, for example packets with common source IP, and limits aggregates with high frequency [Mahajan et al., 2002]. By implementing rate limiting close to the source of congestion the attack is mitigated. This is achieved recursively via the Pushback mechanism which requests upstream routers to undergo rate-limiting of the same aggregate. Pushback originates at a node experiencing congestion where legitimate traffic may be present in large volumes, the approach risks potential collateral damage if legitimate traffic is erroneously grouped into the attacking aggregate [Yau et al., 2005].

Router Throttling

A similar approach, Router Throttling, views the task as a 'resource management problem' [Yau et al., 2005]. It has the dual objective of maintaining the load at the server below maximum capacity whilst ensuring the maximum number of legitimate packets can be serviced. The mechanism is similar to ACC as it also drops packets upstream. Rate limiting requests originate from the server as opposed to congested nodes. *Yau et al.* presented two mechanisms involving a Baseline Throttle and a Fair Throttle that achieved level-k max-min fairness. The baseline approach designates a drop rate, a uniform percentage of traffic that throttling agents must discard in the following interval, which unfairly penalises low traffic flows. The Fair Throttle by contrast uses a forwarding rate which establishes a maximum amount of traffic that can pass through the node until the following interval, the forwarding rate ensures low traffic streams are not unfairly punished [Yau et al., 2005]. Both approaches uses the AIMD control mechanism to maintain the server load between an upper and lower bound. The mechanism generates an instruction every $T_{defender}$ seconds by observing

the load at the server, Alg 1 depicts the AIMD algorithm for the purposes of the Fair Throttle.

The Fair Throttle was demonstrated to mitigate DDoS attacks throughout large simulations of realistic traffic. A mathematical proof of throttling convergence when exposed to a dynamic DDoS attack reduces the risk that such a system can be gamed. The reliance on the server to instruct throttling rates presents a single point of failure, this vulnerability is particularly concerning as the server will be expected to be facing significant congestion at critical points of functionality [Malialis and Kudenko, 2013b].

Algorithm 1: Fair Throttle AIMD Algorithm

```
Input  : Aggregate traffic load at the sever
Output: Uniform forwarding rate for all agents
Result: Maintain server congestion within upper (U_S) and lower (L_S)
        bounds
//Initialise last observed traffic load at the server
p_last = −inf ;
while Active do
    multicast current forwarding rate r_S throttle ;
    monitor traffic arrival rate p for time window w ;
    if p > U_S then
        // Server is congested
        r_S := r_S/2 ;
    else if p < L_S then
        // Server is under capacity
        if p − p_last < ϵ then
            remove throttle ;
            break ;
        else
            // increase the forwarding rate
            p_last := p ;
            r_S := r_S + δ ;
        end
    else
        // Aggregate load is within acceptable boundaries
        continue ;
    end
end
```

MARL

MARL is a Multi-Agent Reinforcement Learning throttling mechanism designed as an alternative to the Fair Throttle [Malialis and Kudenko, 2013b]. The use of Reinforcement Learning allows a decentralised structure where each router has an independent learning agent responsible for setting a throttling rate. The system inherits the Fair Throttle's objectives of avoiding server congestion and minimising the percentage of legitimate packets disrupted. It is similar to the Baseline Throttle as it sets a drop rate of the percentage of traffic to be discarded every $T_{defender}$ seconds as opposed to a forwarding rate like the Fair Throttle. The drop rate does not unfairly punish low traffic streams unlike the Baseline Throttle as MARL offers individualised drop rates through its decentralised structure. The functionality of MARL is elaborated in Sec 3.2.1.

The original MARL mechanism matched or outperformed the Fair Throttle against a series of common attacks in small topologies [Malialis and Kudenko, 2013b]; a variant which incorporated limited hierarchical communication outperformed the Fair Throttle against all simulated attacks in a larger topology [Malialis and Kudenko, 2013a]. In the comparative evaluation, where the agent interval of both mechanisms was set at T=2 seconds, the Fair Throttle observed significant drops in performance against Short Burst attacks whilst MARL demonstrated limit performance degradation. Further investigation is required to confirm whether the Fair Throttle, which uses a forwarding rate, is vulnerable to a Short Burst attack like the results suggest.

The limited communication structure provides a decentralised system that lacks a single point of failure unlike the Fair Throttle. There has been limited research seeking to evaluate the versatility of MARL. As each action only considers the load for the prior $T_{defender}$ seconds, since the last move there is no overlap in observations between successive actions. Consequently there lies a potential vulnerability of a dynamic attack where the router load varies between subsequent actions. The inclusion of the bottleneck, a safety mechanism that throttles congestion at the server, may have been included in the network model to address a vulnerability to variable traffic. The existence of the safety mechanism is a valid assumption, network defences rarely work in isolation, although the reliance of the server to manage congestion may pose a risk in a realistic environment.

Agents receive a global reward to encourage cooperation calculated by the percentage of legitimate traffic serviced. At the conclusion of each step, if the

network avoided congestion that are rewarded by the percentage of legitimate traffic served, otherwise they are punished with a reward of -1. Currently the focus of research has been for network simulations where the reward can be readily calculated. Although no reward mechanism for online learning has been established, a fuzzy reward could be generated through approaches similar to anomaly detection therefore is compatible with online learning.

Since the establishment of MARL there have been significant advances in Reinforcement Learning with particular focus on Deep Reinforcement Learning. Later variants of MARL utilised a condensed hierarchical communication structure to provide agents with awareness of surrounding nodes whilst minimising communication overhead. Current implementations use Linear Function Approximation to interpret the state-space but a non-linear approximator, like Deep Reinforcement Learning, may be able to interpret the condensed state with greater precision. A study in the field of wireless DDoS attacks found Deep Reinforcement learning both increased performance and reduced training speeds [Han et al., 2017], a similar methodology could allow MARL to adapt to the real environment quicker and with greater precision once deployed. Currently all evaluations have employed the bottleneck which represents a contiguous defence system. The provision of a reward of -1 during congestion represents a disparity between evaluation and rewards. If the bottleneck is a valid assumption then it may be of benefit for MARL to encourage cooperation with the bottleneck by rewarding the system on evaluation performance rather than predefined metrics.

2.4 Resilience to Atypical Attacks

Prior literature has provided limited insights of the resilience of the reviewed mechanisms to being gamed or exploited. This is generally not analysed in the publications outside the occasional mathematical proof of convergence seen in the Fair Throttle [Yau et al., 2005]. Arguably this is difficult to quantify and individual analysis of systems is expensive. Commonly when detailing a new mechanism, is evaluated under simulation or emulation through a series of likely attacks [Fayaz et al., 2015, Yang et al., 2005, Mahajan et al., 2002, Yau et al., 2005, Malialis and Kudenko, 2013b]. As attackers become more sophisticated the omission of modelling an intelligent adversary represents a vulnerability in deployed systems.

2.4.1 Realistic Evaluation

A survey of published papers identified a lack of rigour in the evaluation of proposed DDoS defences as they were not simulated under a realistic environment [Mirkovic et al., 2009]. A disparity between the evaluation and actual environment limits the capacity of the researcher to predict how a mechanism will respond in a live environment. Currently the focus of the establishment of common benchmarks has been emulating a realistic network against expected attacks. It is important to consider both the defence's effectiveness against typical attacks as well as their response against atypical attacks designed by an intelligent attacker. Therefore to adequately simulate a realistic evaluation, current benchmarks must be expanded to include the intelligent attacker.

2.4.2 Modelling an Intelligent Adversary

In the field of DDoS attacks, intelligent adversaries have been previously modelled in proposals for defensive systems specifically designed to be resistant to an intelligent attacker. Many of these approaches use Game Theory or Reinforcement Learning to determine the Nash Equilibrium within a theoretical framework, calculating where both the defender and attacker's strategies converge to a point where there is no advantage of either opponent switching strategies [Wu et al., 2010, Yan et al., 2012, Shiva et al., 2010]. These papers operate on a high level theoretical framework that has limited utility as they do not accurately portray a realistic network. The calculation of optimal strategies by an attacker could provide insights of the behaviour to expect from an intelligent adversary and be used to identify vulnerabilities of the defender. These papers have simulated the intelligent adversary through the use of Reinforcement Learning which is compatible with the network simulation used in standard practice evaluation.

2.4.3 Evaluation Through Intelligent Attacks

The application of similar methodology with the focus on the attacker against an established defence would extend current evaluation practices to include the intelligent attacker. Previously the focus of Reinforcement Learning has been to allow a defender to adapt to new environments or attacks. By shifting the focus to the attacker, an attacker that adapts its attack in response to the opposing defender can be emulated. This would provide another metric to assess effectiveness whilst also yielding insights in understanding the resilience against

an intelligent attack. When the focus is on the defender, the researcher must justify the effectiveness and reliability of the attached reward calculator. Evaluation by contrast operates in a closed network simulation where rewards can be readily calculated, hence such restrictions do not apply to the design of the adversary. The design of optimal attacks, approximated through Reinforcement Learning, would extend standard practice evaluation as it would consider a realistic attacker.

We identified a similar aim to ours by *Zhang et al.* which sought to optimise a wireless jamming adversary. The authors noted that prior literature had designed defensive strategies to counter common attacks [Zhang et al., 2015]. Our proposal differs by modelling the most potent attack given both the availability of resources and the network's defender. Reinforcement Learning is compatible with typical evaluation which also uses simulation, would consider the behaviour of defender and thus evaluate the versatility of the system to be gamed.

2.5 Summary

DDoS attacks continue to be a complex problem incorporating many layers of defence. Rate-limiting is an approach used to block incoming streams when we have limited information to discriminate against packets. We have identified Reinforcement Learning as a tool to provide an adaptive defender and provide realistic evaluation as shown in Table 2.1. The goal of the adaptive defender is to provide a defence mechanism that evolves as it is exposed to new networks, user or attack behaviour. Chapter 3 expands on one such system, MARL, which may provide an adaptive rate-limiter. In this review we have identified several vulnerabilities and potential areas of improvement for MARL, therefore there is a need for further evaluation to quantify the effectiveness and versatility of the system.

To properly understand the versatility of a defender we must consider its effectiveness against typical attacks as well as their response against atypical attacks. Research that does model an intelligent adversary, either through Game Theory or Reinforcement Learning, have maintained the focus on the defender typically providing a framework to design models resistant to an intelligent adversary. Thus the open question we seek to answer is *can we learn from the experience from designing intelligent adversaries to develop automated tools for testing the effectiveness of existing defences?* Chapter 4 addresses this challenge

by modelling an intelligent adversary through Deep Reinforcement Learning. To demonstrate the utility of the simulated adversary we use our model to extend the evaluation established in Chapter 3.

	Approach	Description	Benefit	Drawbacks/Challenges
Defence Design	Standard Practice	Designed in closed environment	Reliable Risk Adverse	Limited adaptability to new attacks
	Reinforcement Learning	Continue to learn policy once deployed	Adapts to new environment Adapts to new attacks	Requires reliable rewards Lengthy training Not effective?
	Chapter 3: Can MARL be made a suitable replacement to the Fair Throttle?			
Evaluation Methodology	Standard Practice	Test defender against predefined attacks	Automated Comparative	Does not consider intelligent attacks
	Individual Analysis	Perform investigative analysis on defender	Tests versatility Considers intelligent attack	Not comparative Expensive
	Reinforcement Learning Adversary	Simulate intelligent adversary	Automated Comparative Tests versatility Considers intelligent attack	Unexplored Large dimensionality
	Chapter 4: Can Reinforcement Learning be used to develop automated tools for testing the effectiveness of existing defences?			

Table 2.1: A summary of Reinforcement Learning's contributions to DDoS Defence

Chapter 3

Resistant Defenders

Reinforcement Learning provides a framework to develop decentralised defence mechanisms where agents learn an effective policy through indirect rewards. The approach requires online learning or a demonstration that the learnt policies are effective in the deployed environment. Online learning enables the defender to adapt to changes in the network environment such as new attack or user behaviour. An automated approach to updating network policy simulates the design process defence traditionally undergoes seen in Fig 3.1. An evolving defence mechanism would reduce the need for researchers to investigate and update current defence mechanisms introducing benefits of reduced time and cost for adapting to an attack.

The initial challenge is to establish that a Reinforcement Learning agent can outperform standard practice solutions given sufficient training. This chapter examines the MARL rate-limiter, designed as a Reinforcement Learning alternative to the Fair Throttle [Malialis and Kudenko, 2013b]. MARL has been previously shown to outperform the Fair Throttle against variable traffic [Malialis and Kudenko, 2013b]. We extend prior investigations by considering the versatility of the comparative systems. Furthermore, we introduce two adaptations of MARL that are demonstrated to improve performance against constant traffic.

Our investigation challenges prior studies where we find that the Fair Throttle outperforms MARL against variable traffic. We conclude by identifying a versatility challenge for MARL by examining its reliance of the modelled bottleneck at the server to manage congestion against variable traffic.

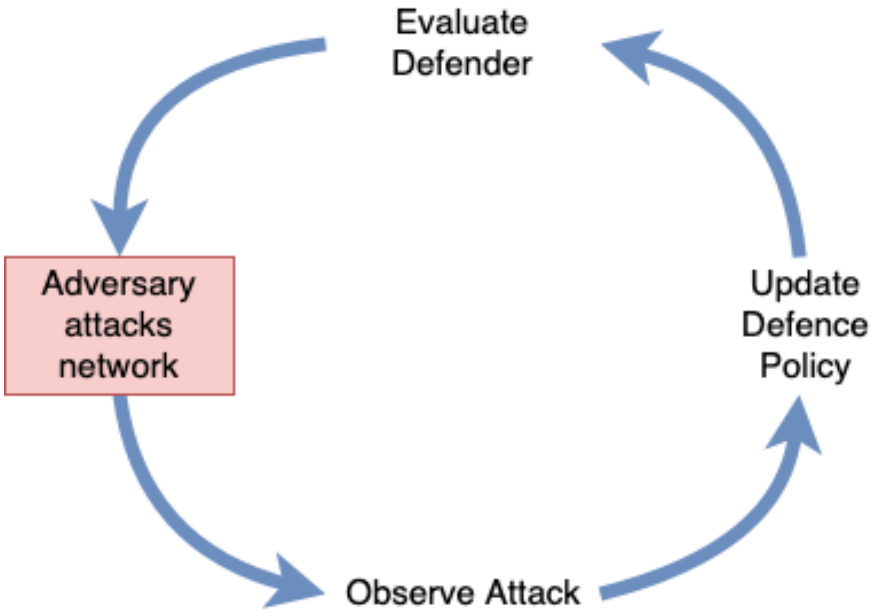


Figure 3.1: Development cycle for DDoS defence design. Can Reinforcement Learning provide an adaptive defender that adapts to the live environment?

3.1 Problem Statement

The goal of the rate-limiter is to avoid server congestion whilst minimising the disruption of legitimate traffic by throttling incoming packets. MARL is an adaptive rate-limiter that learns a throttling policy tailored for the desired network. The establishment of an effective Reinforcement Learning rate-limiter would allow the design of network systems that adapt to new threats or changes in user behaviour. The limited investigation of MARL has left the utility of the rate-limiting approach unclear; in Sec 2.3.3 we identified a potential vulnerability against burst attacks that questions the versatility of MARL. There exists multiple challenges for adaptive rate-limiters, namely the requirements of a reliable reward calculator, the quantification of necessary training, however first it must be demonstrated to outperform current literature once trained. To test whether MARL is a suitable replacement to the Fair Throttle, we investigate under what conditions does the adaptive rate-limiter outperform the Fair Throttle?

3.1.1 Network Model

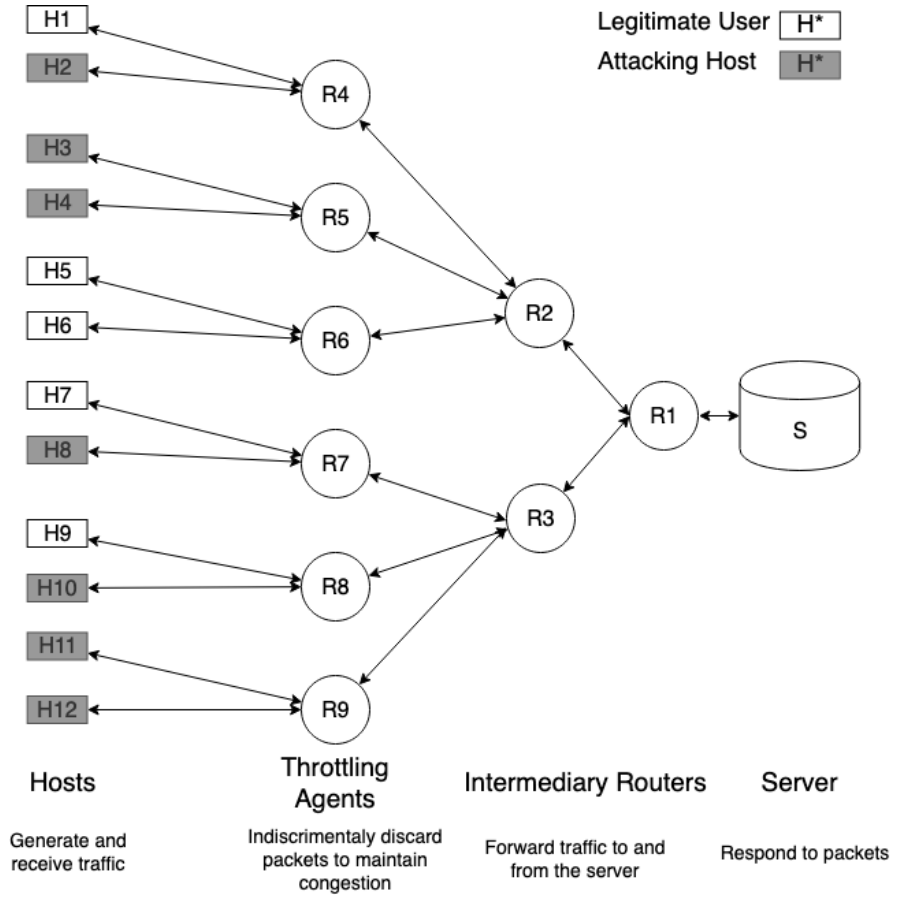


Figure 3.2: The 6 Defender Topology showing the roles of all agents.

Our network is similar to the models used by *Yau et al.* and *Malialis and Kudenko* [Yau et al., 2005, Malialis and Kudenko, 2013b]. The network is a connected graph G=(V,E) where V is the set of nodes and E is a set of bidirectional edges. Nodes may either represent the server (S), internal routers (R_{1-n}) or hosts (H_{1-m}). The network aims to avoid server congestion whilst servicing all legitimate traffic, hosts generate traffic and represent either legitimate users or attackers. Attackers and legitimate users generate traffic at rates r_a and r_l where $r_a >> r_l$, this assumption is valid for typical DDoS attacks, attacks where $r_a \approx r_l$ require larger and more expensive botnets. Network routers can either drop or forward traffic to connecting nodes. The server can service up to

U_S packets per second and is functional while the server's load is below U_S. A subset of routers, designated as 'Throttling Agents', are responsible for dropping traffic to regulate the server load. We adopt a bottleneck, introduced by *Malialis and Kudenko* [Malialis and Kudenko, 2013b], which reduces the load at the server down to U_S during congestion.

3.2 Investigated Defenders

Our evaluation considers five different rate-limiters that are variations of the Fair Throttle and MARL. In this section we provide a summary of the design and functionality of each defender to allow replication of results. This section is structured to provide the functionality of MARL, the two adaptations we introduce and the two Fair Throttle variants. A brief summary of the differences in functionality of each defender is provided later in Table 3.2. We repeat the following definitions which we use to aid the distinctions between agents.

- The *aggregate load* at every node is a measurement of the volume of traffic directed at the server that passes through the node.
- An agent may employ a *monitoring window* to observe the aggregate load at its location for the last $T_{defender}$ seconds.
- Traffic is measured in Mbits/s.
- A *forwarding rate* sets a volume of traffic that can pass through the node directed at the server. A forwarding rate of 0.5 would limit incoming traffic to 0.5 Mbits/s.
- A *drop rate* sets a percentage of traffic that can pass through the node directed at the server. A drop rate of 50% would discard half of the incoming traffic until the next instruction

3.2.1 MARL

We provided detailed analysis of the functionality of MARL that extends what was provided in Sec 2.3.3. By detailing the functionality and parameter settings used we enable future researchers to replicate our results. Furthermore, our adaptations inherit the design of MARL, unless specified our replication of MARL follows the design by the original authors [Malialis and Kudenko, 2013b, Malialis and Kudenko, 2013a, Malialis et al., 2015].

Agent Design

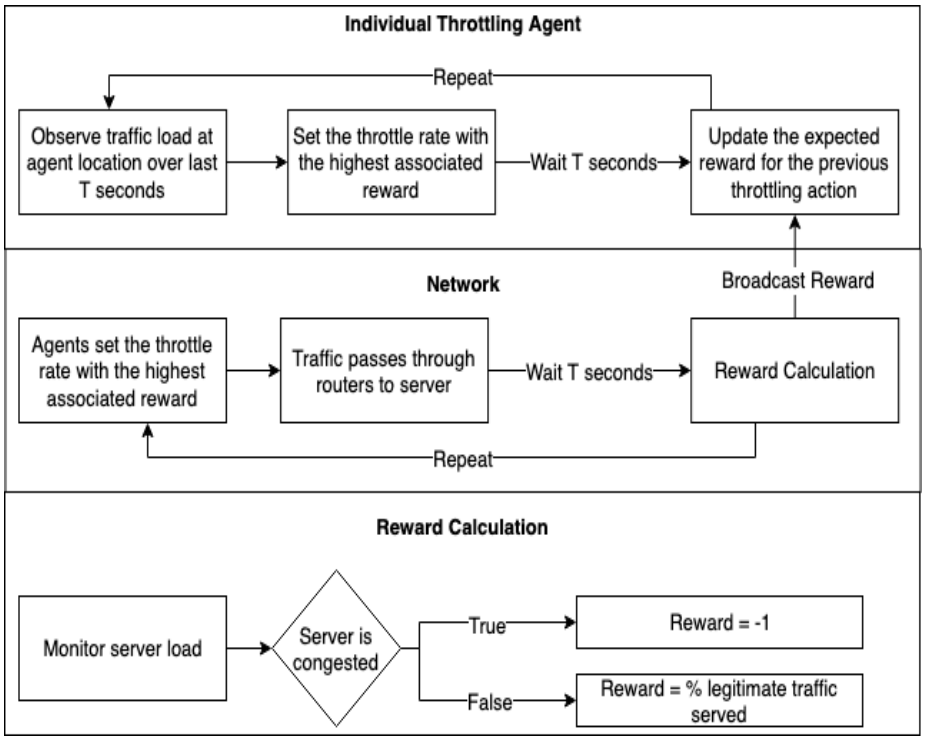


Figure 3.3: Depiction of the learning work flow of MARL. MARL learns through interaction with the network.

MARL inherits the Fair Throttle's dual objectives of maintaining the server load below U_S and servicing all legitimate traffic. It consists of a learning agent installed at every throttling router, Fig 3.3 shows the process of each Throttling Agent learning a cooperative policy. Each agent observes the aggregate load at its location and does not communicate with the server, the decentralised mechanism does not have a single point of failure unlike the Fair Throttle [Malialis and Kudenko, 2013b]. Each step, which is aligned to the monitoring interval, the agent chooses a drop rate that dictates the percentage of incoming traffic is to be discarded until the next step. Agent actions are discretised into intervals of 0.1 where $a \in \{0, 0.1, .., 0.9\}$ representing the percentage of traffic dropped in the following interval. The authors omitted the option to discard all traffic as it is considered unlikely that incoming traffic would consist solely of DDoS traffic, prohibiting all traffic would "facilitate the task of the attacker" [Malialis and

Kudenko, 2013b]. To interpret the continuous state, the system utilises Linear Function Approximation; Tile Coding transforms the continuous state into a discrete set of tiles [Sutton, 2019].

Training

Each agent learns to associate each potential state with a drop rate. The network is trained against constant streams of randomly generated DDoS attacks. At the end of each action a global reward is produced which reflects the dual goals specified and is depicted in Alg 2. Training occurs offline, through simulation prior to deployment, during which we are able to keep track of both legitimate packets sent by Hosts and those received by the server. The learning rate and discount value is set to $\alpha = 0.01$ and $\gamma = 0$. MARL is reactive system where the attacker controls the subsequent state, therefore the system does not incorporate future rewards providing the following update rule:

$$Q(s, a) <\text{-}\ Q(s, a) + \alpha[r - Q(s, a)]$$

During simulation the MARL agent undergoes 120,000 episodes of 60 seconds length including 20,000 episodes of pretraining followed by 60,000 episodes where the exploration rate is linearly decreased from 1 to 0.

Algorithm 2: MARL Global Reward Calculator

Input : Aggregate load at server over the last $T_{defender}$ seconds
Output: Global Reward for all learning agents
Result: Train agents to avoid server congestion and maximise number of legitimate packets served

```
while Active do
    monitor server traffic loadRouter_server over T_defender seconds ;
    if loadserver > ServerCapacity then
        // Punish all agents
        reward = -1 ;
    else
        // Reward for legal traffic serviced
        reward = legitimateLoad_server/legitmate_sent;
    multicast reward to all agents
```

3.2.2 MARL Adaptations

Having established the fundamental model we now provide our two adaptations. The purpose of these improvements is to increase the performance by minimising disruption of legitimate packets.

Alternative Reward Structure

We provide a new reward structure that address the discrepancy between rewards and evaluation. The adapted agent, MARL-A, differs from MARL by the new reward mechanism where each agent is rewarded by the percentage of traffic serviced even during congestion.

The original reward mechanism punishes agents when they fail to avoid congestion however prior evaluation methodology has allowed a bottleneck to reduce congestion at the server. Let us consider a scenario where all agents have declined to throttle a flash crowd consisting of legitimate traffic equalling $1.2U_S$. During evaluation the bottleneck would drop 9% of traffic resulting in a metric of 0.83 whilst the prior structure would see each agent punished with a score of -1. It is unlikely that the defender operates in isolation and therefore we risk over-throttling due to the considerable punishment associated with congestion. Cooperation with pre-existing defences, which can be difficult to design in Control Mechanisms like the Fair Throttle, are encouraged through appropriate reward structures.

Deep Reinforcement Learning

Our second approach to improve performance, MARL-DH, considers the use of Deep Reinforcement Learning to aid the agents for making optimal decisions. Previously we have considered a limited state-space that consists of a single variable, increasing the state-space to include traffic loads of nearby routers allows the agent to make a more informed decision. Later investigations have incorporated a hierarchical structure, where the state consists of the aggregate load of two intermediate routers in the direction of the server to mitigate scalability challenges [Malialis and Kudenko, 2013a]. MARL-DH inherits the hierarchical communication structure, our new reward calculator and learns through DDQN described in Sec 2.2.3.

The communication structure represents a condense summary of the network, In the topology depicted in Fig 3.2 in Sec 2.3.3, the Throttling Agent at

R4 would observe the state space <R4, R2, R1>, the expanded state expresses a non-linear relation with the aggregate load of nearby throttling agents. If the condensed state-space exhibits non-linear relationships, a neural network would be able to interpret relationships between states to a greater extent than Linear Function Approximation. Furthermore, a similar study seeking to mitigate wireless DDoS attacks found Deep Reinforcement to improve performance and reduce training required [Han et al., 2017]. DDQN may provide a tool to increase performance whilst decreasing training required enhancing the adaptability of MARL.

To localise the improvement due to the use of Deep Reinforcement Learning as opposed to an extended state-space, we have modelled an identical Function Approximation agent which we name MARL-FH. Both MARL-DH and MARL-FH undergo 20,000 episodes of pretraining followed by 60,000 episodes of linearly decreasing exploration. As our focus was on performance rather than training speed we have allowed the Function Approximation mechanism to have extended training post exploration, the total number of training episodes for MARL-DH and MARL-FH is 150,000 and 350,000. Both agents use the new reward mechanism proposed. Our simulation incorporates communication delays where agents must receive the entire state-space before making an action, a $20ms$ delay of throttling decisions models the communication delay for the furtherest state incorporated.

3.2.3 Comparative Baseline

Our evaluation compares the MARL variants to the Fair Throttle described in Sec 2.3.3. The Fair Throttle differs from MARL as it sets a through rate which seeks to limit the volume of traffic passing through each router. Throttling decisions determined by the AIMD algorithm, depicted in Alg 1 under Sec 2.3.3, which maintains incoming traffic within a lower and upper bound. We provide two implementations that differ due to the throttling mechanism. The Fair Throttle sets a 'forwarding rate' which limits the volume of traffic that can pass through the router with any excess packets dropped, therefore if the incoming load was to drop below this threshold, the router would allow all traffic to pass. The functionality is similar to the Traffic Policing mechanism supported by Cisco routers that support 'Cisco Express Forwarding' [Cisco, 2017]. The forwarding rate is different to the drop rate used by MARL which is limited in its capacity to adapt to variable traffic. Our variant, the Fixed Throttle,

sets a 'drop rate' that discards a specified percentage of incoming traffic, this is equivalent to the forwarding rate assuming constant traffic. For example, provided a through rate of Mbits/s and experiencing a load of 10Mbits/s the agent would then discard 50% incoming traffic until the next step.

The Fixed Throttle shares the throttle mechanism with MARL and struggles with variable traffic. Table 3.1 demonstrates the varying performance of the Fair and Fixed Throttle where we consider a single throttling agent topology with the inclusion of a bottleneck against a variable DDoS attack. Here the AIMD algorithm sets a through rate to maintain traffic between L=6 and U=10, traffic volume is measured in Mbit/s. Both implementations set identical through rates from steps 1 to 9 however the Fixed Throttle continues to drop incoming packets for the following step after the attack stops. By the tenth step the Fair Throttle has serviced 23% more legitimate packets whilst the drop rate has resulted in the Fixed Throttle setting a through rate below the lower boundary demonstrating a clear over throttle.

	Step	1	2	3	4	5	6	7	8	9	10	Total
Traffic	DDoS Traffic	20	20	0	4	20	20	0	4	20	20	
	Legal Traffic	1	1	1	1	1	1	1	1	1	1	10
	Load R_1	21	21	1	5	21	21	1	5	21	21	
Fair	Through Rate	inf	8	9	10	11	5.5	6.5	7.5	8.5	8.5	
	Server Load	21	8	1	5	11	5.5	1	5	8.5	8.5	
	Legal Served	0.5	0.4	1	1	0.5	0.3	1	1	0.4	0.4	6.4
Fixed	Through Rate	inf	8	9	10	11	5.5	6.5	7.5	8.5	4.3	
	Server Load	21	8	0.4	5	21	5.5	0.3	5	21	4.3	
	Legal Served	0.5	0.4	0.4	1	0.5	0.3	0.3	1	0.5	0.2	5

Table 3.1: Comparison of legitimate packets served by the Fair Throttle and Fixed Throttle. The Fixed Throttle, which approximates the forwarding rate, is unable to adapt to variable traffic as effectively as the Fair Throttle.

3.3 Simulation Results

In each episode we partition Hosts into attackers and legitimate users, a Host has a probability of 0.6 of being a legitimate user. Each legitimate user and attacker produces traffic at rates $r_l \in \{0.05, 1\}$, $r_a \in \{2.5, 6\}$. We set both the monitoring window and the time between agent actions at $T_{attacker} = 2$ seconds. A $10ms$ delay exists between adjacent nodes and to simulate communication delays for throttling agents that require communication. All edges have infinite capacity

outside the bottleneck S-R_1 capped at U_S. During training all agents undergo both weak and effective attacks, our evaluation however involves only effective attacks where the total packet capacity exceeds $1.2U_S$. All agents train on the same topology that they are to evaluated on. Both training and evaluation were performed on a network emulator written in Python on the SPARTAN HPC machine [Meade et al., 2017].

3.3.1 Evaluation Criteria

The mean percentage of legitimate traffic serviced by each defender is measured. We aim to identify which rate-limiters are able to service a greater number of legitimate packets given different DDoS attacks. Evaluation lasts for 120 seconds with the attack beginning at t=10 and finishing at t=110. We generated 500 random episodes of attack distributions under the constraints set out in Section 3.3. We test defenders against a series of typical attacks [Mirkovic and Reiher, 2004], the independent variables are limited to the defender used and the attack behaviour however randomised training of MARL can lead to differing performance for MARL agents. Therefore all evaluations are repeated 10 times using the same 500 episodes with independent random training, we display the overall mean packets served as well as the variation of the mean over repetitions. The variance of the mean over repetitions reflects the likelihood of achieving the same average performance given random training of the learning agent. Claims of significance in this thesis have be determined through a Paired Sample T-Test that used all episodes. To aid the reader we have provided a summary of defenders and attacks used:

Name	Agent	Throttle Mechanism	State Space	Notes
MARL	F.A.	Drop Rate	Local Router	Punish Congestion
MARL-A	F.A.	Drop Rate	Local Router	Reward Performance
MARL-FH	F.A.	Drop Rate	Three Routers	Reward Performance
MARL-DH	DDQN	Drop Rate	Three Routers	Reward Performance
Fixed Throttle	AIMD	Drop Rate	Load at Server	Centralised
Fair Throttle	AIMD	Forwarding Rate	Load at Server	Centralised

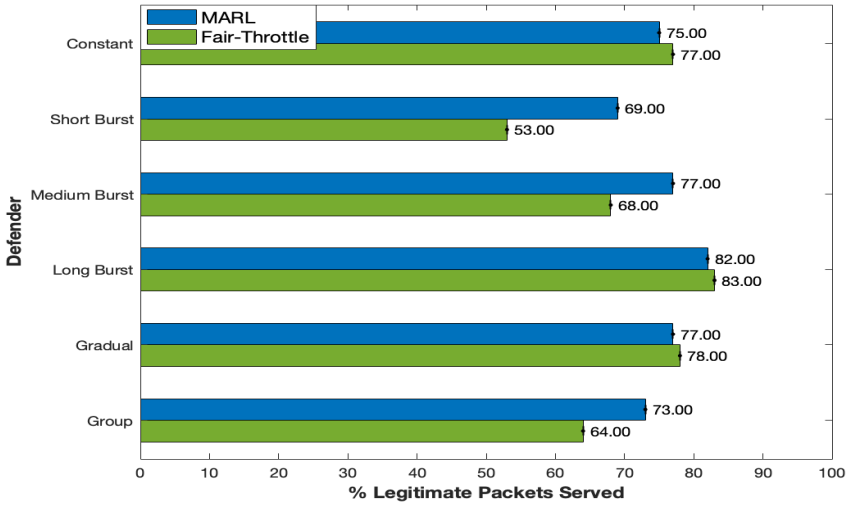
Table 3.2: Summary of different defenders evaluated. F.A. (Function Approximation) and DDQN are Reinforcement Learning approaches.

Attack Name	**Description**
Constant Attack	Attackers maintain constant traffic at the maximum rate
Short Burst	Attackers switch between maximum and zero capacity over a two second interval
Medium Burst	Burst attack with four second interval
Large Burst	Burst attack with ten second interval
Gradual Attack	Attackers incrementally increase the DDoS traffic produced from 0 reaching their adversarial potential halfway through the attack where it remains at maximum volume
Group Attack	We split the attacker into two groups, each engaging in a different attack pattern. Each group can either be Constant Attack, Short Burst or Long Burst

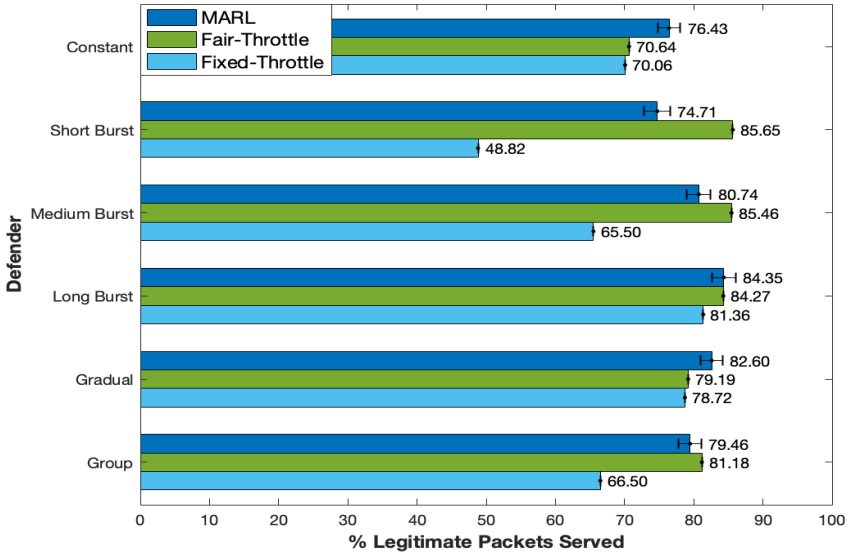
Table 3.3: Evaluated Attacks for Chapter 3.

3.3.2 3 Agent Topology

We replicate the original comparison between MARL and the Fair Throttle to address potential variations of the experiment compared to the prior work. The three agent topology has six asymmetrically placed hosts from MARL's original publication [Malialis and Kudenko, 2013b]. All network and agent parameters listed in the publication are adopted in this experiment, the lower and upper bound of the server are $L_S = 6$, $U_S = 8$. The purpose of the experiment is to provide contextual information how our simulation compares to the prior implementation thus we have restricted evaluated agents to the Fair Throttle and



(a) Results from the original publication [Malialis and Kudenko, 2013b]. We were unable to source the standard deviation from the original work.



(b) The replicated experiment shows clear distinction in the performance of the Fair Throttle. The Fixed Throttle behaves similarly to the Fair Throttle by *Malialis and Kudenko*.

Figure 3.4: Comparison of prior work measuring the mean percentage of traffic served in three agent topology. It is found that the replicated Fair Throttle differs from the design by *Malialis and Kudenko*.

the original MARL agent. This experiment used the original training parameters for MARL which involve 50,000 episodes of linearly decreasing exploration where ϵ is decreased from 0.4 to 0.0 followed by 12,500 episodes of exploitation, the learning rate has been set at $\alpha = 0.1$.

Fig 3.4 displays our replicated results as well as the original results which we sourced from the original publication, we were unable to source the standard deviation from the paper [Malialis and Kudenko, 2013b]. Our MARL results align with the original replication, slight discrepancy of results is expected as it is dependent on the training initialisation and the strength of the DDoS attacks during evaluation. There exists a clear inconsistency with the performance of the Fair Throttle in our evaluation and prior work. Relative to the Constant Attack, our Fair Throttle improves performance against the Short Burst, the Fair Throttle by *Malialis and Kudenko* as well as our Fixed Throttle by comparison suffers a significant decrease in performance. It is our belief that the prior evaluation utilised a mechanism similar to the Fixed Throttle that used a drop rate like MARL, this however would not be a truthful replication of the ideal Fair Throttle which uses a forwarding rate through a virtual leaky bucket. This calls into question the validity of prior comparisons hence the need for our comparative investigation if MARL outperforms the Fair Throttle.

3.3.3 6 Agent Topology

Having established that our MARL agent reflects the original publication we now consider the adaptation listed in listed Sec 3.3.1. We have omitted the Fixed Throttle from our results as it was strictly outperformed by the Fair Throttle. This experiment uses a topology of 6 throttling agents where there are 12 hosts, the server capacity increases to $U_S = 14$ with a lower boundary of $L_S = 10$. Fig 3.2 provides a visual representation of the topology.

MARL fails to outperform the Fair Throttle in any attack in the six agent topology. The Fair Throttle significantly outperforms all other defenders in the Short and Medium Burst suggesting that the limited observation window observed by MARL is vulnerable to variable traffic. The new reward mechanism, aligned with performance, compares favourably against all attacks outside the Short-Burst where MARL's tendency to over-throttle was advantageous. The new reward mechanism learnt to reduce congestion and coordinate with the bottleneck demonstrating the reward mechanism does not require negative shaping to discourage congestion. We expect the utility of the bottleneck to decrease

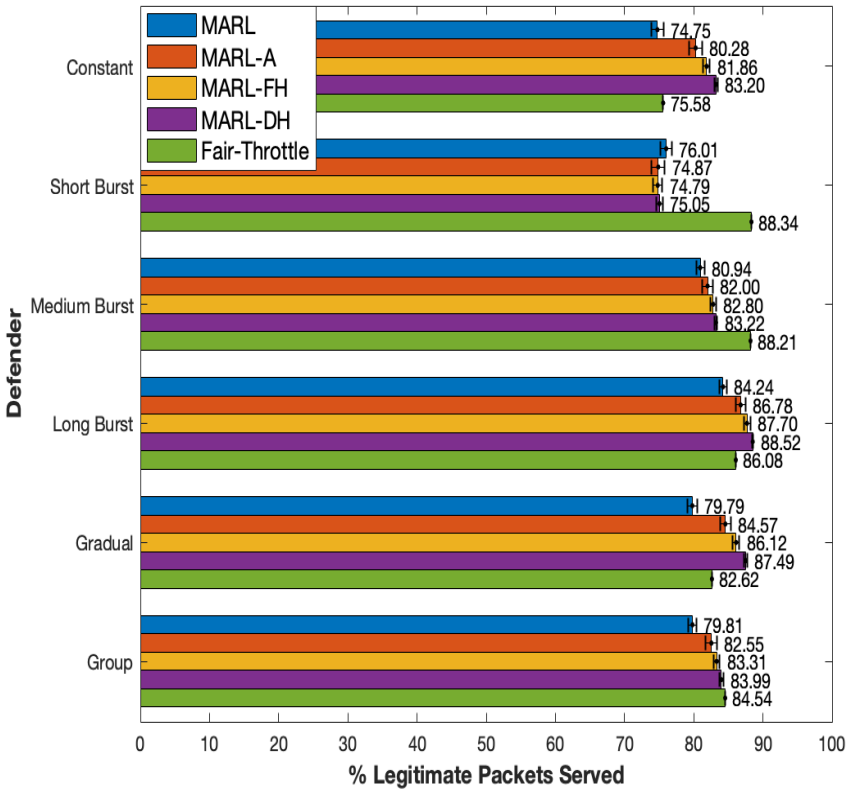


Figure 3.5: The legitimate traffic served is measured in a six agent topology. A summary of defenders and attacks used can be found in Sec 3.3.1.

for larger simulations, however this would be considered by the reward during training. Further, this suggests that when the reward is calculated from performance, rather than arbitrary goals, the agent can learn to coordinate with concurrent defence mechanisms.

The extended state-space used by both MARL-FH and MARL-DH further improved performance for all attacks outside the Short-Burst. MARL-DH which introduced Deep Reinforcement Learning either matched or outperformed the Function Approximation comparative agent MARL-FH in every attack. In every attack either the Fair Throttle or MARL-DH mitigated the attack most effectively. Typically, a DDoS attack would employ a constant stream [Mirkovic and Reiher, 2004] where MARL-DH proved advantageous however the results

show that the Fair Throttle to be resistant to variable traffic where performance was observed to increase.

3.3.4 Removing the Bottleneck

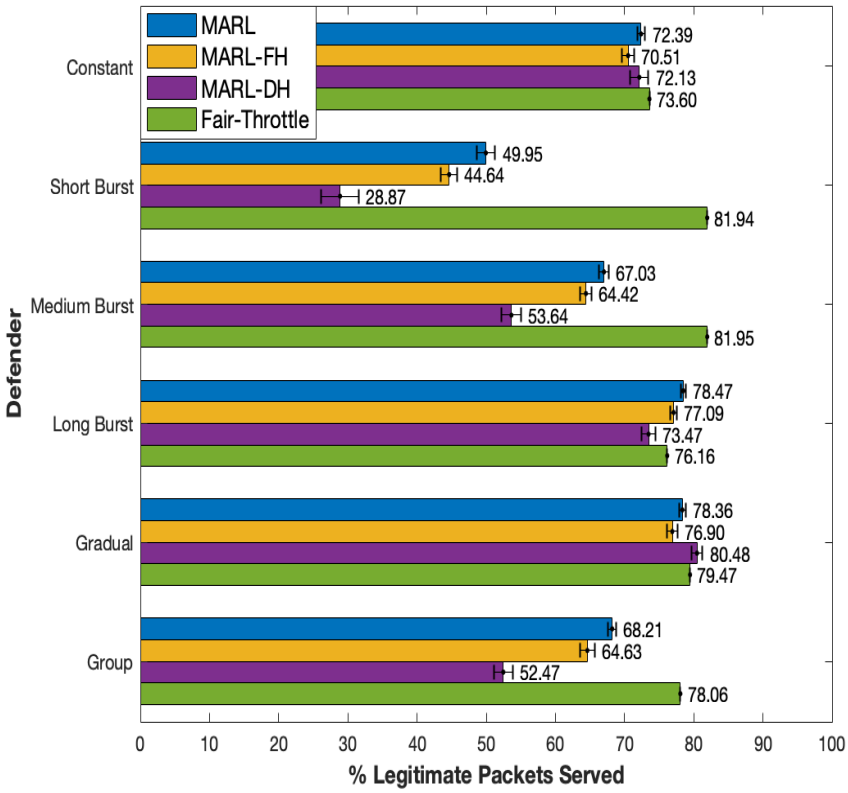


Figure 3.6: The versatility of the defenders is examined by removing the server's capacity to mitigate congestion. The legitimate packets served is measured where no packets are serviced for one second intervals of server overload.

Our last experiment in this chapter investigates the versatility of the proposed defenders and compares how MARL agents and the Fair Throttle achieve their first objective of maintaining the server load below capacity. Prior experiments up until now have reduced server traffic down to U_S through the use of a bottleneck, described as a "safety mechanism" [Malialis and Kudenko, 2013a]. The introduction of a safety mechanism is reasonable in many environments as

we would expect concurrent safety measures in typical servers; our investigations showed considerable performance improvements when cooperation with parallel mechanisms encouraged. The existence of the bottleneck limits our ability to understand the utility provided by the throttling agents alone where in larger topologies we can not rely on safety mechanisms of the server. Over-reliance on the server to maintain congestion risking server failure.

To assess an agent's ability to avoid server congestion we repeat our 6 Agent experiment and assign a value of 0 for one-second intervals where the server load exceeds U_S. This evaluation metric is similar to the original reward mechanism is a value of -1 was assigned for intervals of server congestion. The new reward mechanism are inappropriate and are therefore not included in this evaluation, we remodel the hierarchical rate-limiters to utilise the original reward mechanism.

In Fig 3.3.4 the Fair Throttle matches or outperforms all defenders in every attack outside the Large Burst. Against the Short-Burst, MARL suffers a 22% decrease in packets served despite the attack incorporating half the adversarial DDoS packets than the Constant-Attack. With the bottleneck, MARL variants exhibited limited degradation against Short-Bursts demonstrating that past performance was reliant on the existence of the bottleneck. Therefore MARL, in the mechanism proposed by *Malialis and Kudenko*, is not an appropriate tool for avoiding server congestion unlike the Fair Throttle.

3.4 Discussion and Conclusion

3.4.1 Analysing MARL

In Sec 3.3.4 we observed an inability for MARL to mitigate variable traffic aligned with the monitoring window. This would suggest that past performance against variable traffic has been reliant on the existence of the bottleneck posing a versatility risk for MARL. An obvious approach to respond to variable traffic could be to expose the MARL agents to dynamic attack patterns during training, such an approach however would not address the core vulnerabilities of the system.

A cause of the vulnerability is due to the limited observation window for decision making and the use of a drop rate as opposed to a forwarding rate. The limited state consists of a state-space that consists of the aggregate load for the last $T_{defender}$ seconds. Consider the Short Burst which oscillates traffic

rates emitted aligned with the monitoring window. After the off-phase the router having experienced a high load would aggressively throttle legitimate traffic. Following a step of low traffic the agent would relax the defences and lower the throttle allowing the attacker to bypass the throttling agent.

The short observation window allowed MARL to avoid the need to converge to a throttle rate like the Fair Throttle [Malialis and Kudenko, 2013b] however this risks the mechanism overreacting by limiting past considerations to a solitary state-space. The limited state-space is incompatible with training against variable traffic as it would fail to differentiate a Burst Attack from constant stream of traffic, therefore it may design a policy that over-throttles low traffic streams due to the potential of an impeding high burst. Our adaptations showed that Reinforcement Learning can outperform the Fair Throttle against streams of constant traffic, for MARL to outperform the Fair Throttle against variable traffic there must be an expansion to the state-space that considers the rate traffic is changing as well as the load. We leave further expansions of MARL to future research.

3.4.2 Conclusion

This chapter reinvestigated the prior comparison of the proposed MARL agent [Malialis and Kudenko, 2013b] against the established Fair Throttle [Yau et al., 2005], further we contributed two proposals designed to allow optimal throttling decisions. By observing each agent without the bottleneck we identified MARL's reliance on the server to manage congestion during variable traffic. We further explained a vulnerability shared by the Fixed Throttle and MARL when a fixed drop rate approximated a forwarding rate. From this we conclude that the MARL agent, in its current implementation, is not a suitable tool to avoid server congestion.

In simulations that included a safety mechanism at the server, MARL failed to match the performance of the Fair Throttle against Short Burst attacks. This finding was contrary to the comparison by *Malialis and Kudenko* where the observed Fair Throttle results were similar to our Fixed Throttle suggesting the prior work set a drop rate rather than a forwarding rate. Therefore our work contributed to prior literature by comparing MARL to an ideal Fair Throttle. The discrepancy of the two Fair Throttles highlights the importance of the appropriate throttling mechanism. Our analysis considered small topologies, the evaluation without the bottleneck demonstrated MARL's inability to

manage congestion against variable traffic. As the size of the topology increases the utility of the bottleneck decreases, thus we would expect MARL to scale unsatisfactorily compared to the Fair Throttle.

All of the proposed adjustments improved performance of the agent for constant traffic. By aligning rewards to performance, MARL coordinated with the bottleneck to limit over-throttling. The performance increase associated by the introduction of Deep Reinforcement Learning demonstrated that a non-linear neural network was able to interpret the state-space with greater clarity than the Tile Coding approach. Deep Reinforcement Learning has been shown to be more scalable for state-spaces in Attack Reaction simulations than Function Approximation [Han et al., 2017]. Therefore further expansions to MARL that may incorporate a larger state-space may benefit from such an approach.

There are two distinct differences between MARL and the Fair Throttle, relating to the decision agent and the throttling mechanism. Our simulations showed that a Reinforcement Learning approach can outperform the Fair Throttle in constant traffic however struggled against variable traffic. Decreasing the interval between successive actions would be expensive in an environment where we cannot afford significant overhead, it is our suggestion that further work in improving MARL should consider using a forwarding rate as opposed to a drop rate.

In conclusion our proposed improvements saw MARL outperform the Fair Throttle during constant traffic, however in simulations with large discrepancy of the adversarial traffic rate the Fair Throttle serviced a greater number of legitimate traffic. Having established MARL to be reliant on the server to mange Burst Attacks we establish a versatility risk with MARL and conclude, that in its current implementation, it is not a viable rate limiter. Our evaluation consisted of a set of predefined likely attacks, Chapter 4 considers how to broaden evaluation to consider an intelligent attacker.

Chapter 4

Modelling an Intelligent DDoS Adversary

In Chapter 3 we evaluated variations of the Fair Throttle and MARL defender. Our investigation followed standard practice, where each mechanism was subject to a series of expected attacks. It was demonstrated that other than an ideal environment, once the bottleneck was removed, MARL fails to control congestion. The previous chapter can be considered a cautionary tale of the need to challenge assumptions that may not be necessarily true in a live environment. The difficulty of predicting how an adversary will behave has seen prior evaluations only test against observed attacks, commonly a new DDoS proposal is evaluated through simulation against data sets of typical DDoS traffic [Yang et al., 2005, Nguyen and Choi, 2010, Javaid et al., 2016] or a series of likely attacks [Fayaz et al., 2015, Mahajan et al., 2002, Yau et al., 2005, Malialis and Kudenko, 2013b]. The focus of this chapter is to tackle assumptions of attack behaviour by simulating the intelligent adversary.

In this chapter we propose the use of Deep Reinforcement Learning to model an intelligent adversary which learns attack patterns based on a defender's vulnerabilities. Our Intelligent DDoS Adversary (IDA) provides a cost effective evaluation tool to determine how a rate-limiting DDoS defender will perform against an intelligent attack. The approach contributes to standard practice by providing an additional metric that informs the researcher of the effectiveness of the defender against intelligent attacks. We demonstrate IDA against a series of rate-limiting defenders, where the defender is exposed to both intelli-

gent and typical attacks. We found that IDA outperformed the most damaging standard attack against all tested defenders in different network topologies. We then analyse the attack patterns generated by IDA to illustrate an undiscovered vulnerability against the MARL defender.

4.1 Problem Statement

A DDoS defence mechanism should be able to reduce DDoS traffic to a sufficient level to alleviate disruption for legitimate users. The cost of a successful DDoS attack to the victim requires the proposed mechanism to be evaluated prior to deployment. Evaluation must be realistic to assess both the expected performance and associated risks of the mechanism. Ideally, evaluation should consider a realistic network as well as potential behaviour by both legitimate users and the adversary. Adversarial modelling should include both expected and atypical attacks [Mirkovic et al., 2006]. The need for realistic emulation for evaluation is an established research challenge [Mirkovic et al., 2009] however the absence of an automated intelligent attacker neglects to consider a realistic adversary. Proposed evaluation test beds, like the work by *Kotenko and Ulanov* [Kotenko and Ulanov, 2014], model realistic networks and agent interactions but provide limited agency for the adversary to design their attack. Previously, automated modelling of an adversary has been limited due to the computational challenge of understanding both the functionality of the defender and the availability of resources.

Recent advances in Deep Reinforcement Learning provide an opportunity to approximate an attacker despite the large non-linear state-space required to be interpreted. To aid the design of resistant defences we propose extending the evaluation of current infrastructure to include an intelligent adversary, modelled through Deep Reinforcement Learning, which approximates the ideal attack policy given the availability of DDoS resources against the defender.

4.1.1 Extension to Network Model

We extend the network model introduced in Sec 3.1.1 by introducing an intelligent adversary that coordinates the attack. The extension considers the addition of an adversary modelled through a set of Handlers that sends commands to attacking hosts. Handlers are detached from the network and direct attacking hosts via broadcasts. Our adversary is similar to the model designed

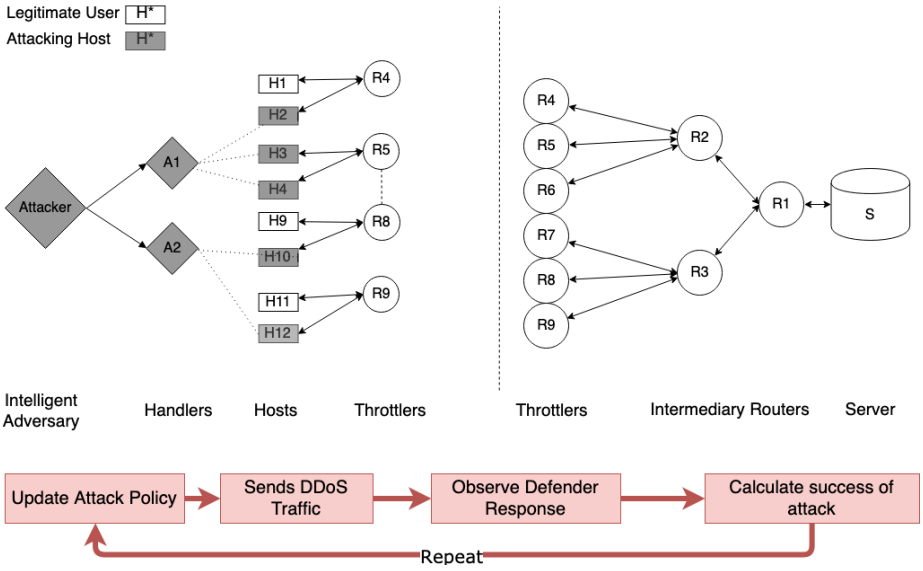


Figure 4.1: Generalised illustration of the adversary's interaction with the network defender. The intelligent adversary learns attack patterns to optimise network disruption.

by *Lin and Tseng*, displayed in Fig 2.1 under Sec 2.1, where attacking hosts are delegated to one of many Handlers [Lin and Tseng, 2004]. The rate-limiters that are investigated respond to the volume of packets as opposed to the contents, thus we restrict the agency of the adversary to consider the quantity of traffic produced. After the attack is initiated, each Handler directs the volume of traffic to be emitted by their respective Hosts. Handlers direct hosts every $T_{attacker}$ seconds with the shared goal of disrupting legal traffic. The network's agents set throttling rates every $T_{defender}$ seconds to achieve the contrary goal of maximising the number of legitimate packets being serviced. As the focus of this section is the design of optimal attacks we relax constraints on the defender to allow a subset of evaluations to consider scenarios where the defender has more actions than the adversary.

4.2 Intelligent DDoS Adversary (IDA)

This section details the design of our adversary, IDA, whose goal is to minimise the number of legitimate packets reaching the server. By simulating the adversary, through Reinforcement Learning, we provide a generic adversary capable

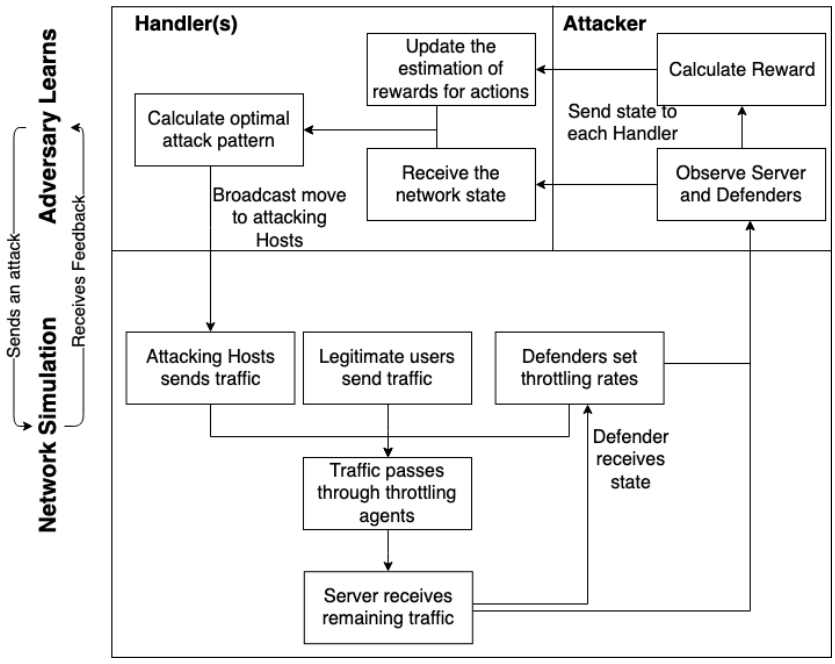


Figure 4.2: Detailed illustration of the interactions of all actors in the learning of optimal attack patterns.

of analysing proposed defenders.

The adversary trains through simulation against a defence policy. After each action the adversary observes the success of the last move and the corresponding action by the defender. Over time, given a defence policy, each Handler learns an attack policy that exploits vulnerabilities in the defender. Fig 4.2 illustrates the interaction between the adversary and the network where the adversary formulates a policy.

The overall goal of our work is to enable defenders to be resistant against an intelligent attack by providing realistic evaluation. To design an effective adversary we must adopt the contrary goal, the purpose of the adversary is to identify attack patterns that minimise the number of legitimate packets serviced by the network. For the intelligent adversary to be of benefit to defence evaluation it should either match or outperform all standard attacks when provided with the same adversarial resources.

We define the adversarial potential as the total DDoS traffic that a unit controls, each attacking Host has an adversarial potential of their generating capacity whilst a Handler's adversarial potential would be the summation of

their Host's potentials.

4.2.1 Structure

The adversary is similar to the DDoS architecture shown in Fig 2.1. The adversary broadcasts a global state to all Handlers every $T_{attacker}$ seconds; Handlers determine the rate of traffic to be produced by their Hosts. IDA adopts a semi-decentralised structure, where a Reinforcement Learning agent is installed at every Handler, to mitigate the size of the action-state combinations which grows exponentially as the number of Handlers increases. Each Handler utilises Deep Reinforcement Learning, specifically DDQN which was described in Sec 2.2.3, to interpret the state space to make their corresponding action. An increase in the number of Hosts allows individualised attack patterns where Handlers can employ differing attacks. A key consideration for the adversary is to balance the competing advantageous of individualised attack patterns through additional Handlers with the typical challenges arising from simultaneous learners such as the difficulties associated with learning in a non-stationary environment [Buşoniu et al., 2010].

4.2.2 Action Space

Each Handler commits to the fraction of the Host's adversarial potential produced in the following interval, Handlers generate a move every $T_{attacker}$ seconds. For example an action of 0.6 would result in each associated Host generating 60% of their adversarial potential. To reduce the size of the Q-Table, the action space is discretised into intervals of 0.1, $a \in \{0.0, 0.1, ..., 0.9, 1.0\}$. As each Handler has 11 possible moves, for every nth handler there exists 11^n potential action combinations.

4.2.3 State Space

Each step the adversary is provided a representation of the network, referred as the state, to aid in the design of attack patterns that maximise network disruption. To maximise the potency of the attack we have chosen not to restrict information that the adversary would typically estimate such as the success of prior moves. Handlers make moves simultaneously having received an identical state, although there is no direct communication between Handlers, they learn an attack policy that indirectly considers the likely moves by other Handlers.

Our experiments utilise a global state that reflects the entire network to encourage coordination. The size of the global state was a key consideration for the use of DDQN as opposed to traditional Function Approximation. Deep Reinforcement Learning has been demonstrated to outperform humans in video games that required interpreting large states [Van Hasselt et al., 2016]. A global state is not scalable for larger and more complex environments which would require the provision of a state-space tailored for individual Handlers.

Our state includes the traffic potential and location of all hosts. It includes a snapshot of the current environment by providing the last 6 seconds of actions by both Handlers and Defenders as well as the server's aggregate load. Our state-space provides the necessary information for the Handler to consider both the available resources and counter the defender's response.

4.2.4 Reward Structure

A global reward is utilised where each agent is equally rewarded for the combined disruption of the network. The sole objective of each Host is to disrupt legitimate service, therefore we frame the design of the adversary as a fully cooperative problem. The use of a global reward avoids potential selfish behaviour that can arise from local rewards [Panait and Luke, 2005]. Let l be the percentage of legal traffic that was serviced in the prior $T_{attacker}$ time step, our adversarial reward is calculated by $r_{attacker}(l) = 1 - l$. The state and reward structure enables the Handler to consider future rewards. The absence of any negative reward encourages each Handler to maximise the disruption throughout the entire episode even if this would require committing to a suboptimal short-term action to position the defender in a vulnerable state.

4.3 Experimental Design

To demonstrate the versatility of our adversary we repeat our experiment against five defenders. We extend the network settings used in Sec 3.3 where all traffic rates and server capacities are measured in Mbit/s. The network has a static defence policy consistent during both evaluation and training of the adversary. Each legitimate user and adversary produces traffic at rates $r_l \in \{0.05, 1\}$, $r_a \in \{2.5, 6\}$. Attacking hosts are delegated to the closest Handler where we set the adversary's interval $T_{attacker} = 2$ seconds.
We model a $10ms$ delay between adjacent nodes, edges have infinite capacity

outside the bottleneck S-R_1 capped at U_S. Our training and evaluation of the adversary involves only effective attacks where the total packet capacity exceeds $1.2U_S$.

4.3.1 Training

In each experiment we train our adversary against a static defender that has either pre-trained or uses an algorithmic mechanism. Our DDQN adversary was trained over 350,000 randomly generated episodes of 60 seconds length where the attack is initiated at $T = 10$. There were 50,000 episodes of pre-training followed by 150,000 episodes of linearly decreasing exploration. Our learning rate and discount value was set at $\alpha = 0.005$ and $\delta = 0.8$. The training of IDA differs from MARL by considering only effective attacks, MARL by contrast was trained against both effective and ineffective attacks. The differing training procedures consider the role of the actor, the defender must learn the appropriate throttling rate for all potential traffic volumes to avoid unnecessary throttling; during evaluation the adversary will always be provided an adversarial load sufficient to achieve congestion.

Fig 4.3 displays the percentage of legitimate traffic serviced throughout the training of IDA against the Fair Throttle for a 3-defender topology averaged over 10 repetitions. Over time the adversary learns a policy that minimises the number of legitimate packets serviced. The variation of available resources each episode is reflected in the broad range of effectiveness after the adversary converged on a policy.

4.4 Simulation Results

IDA is designed to be compatible with common evaluation practices, it should be considered one of many attacks each defender is simulated against. Therefore we extend the simulation procedure enacted in Chapter 3 where a defender is now subject to a series of likely and intelligent attacks. Evaluations consist of 500 episodes which we repeat a total of 10 times, episodes from simulations used in network topologies from Chapter 3 are recycled. Therefore episodes last for 120 seconds with the attack beginning at t=10 and finishing at t=110.

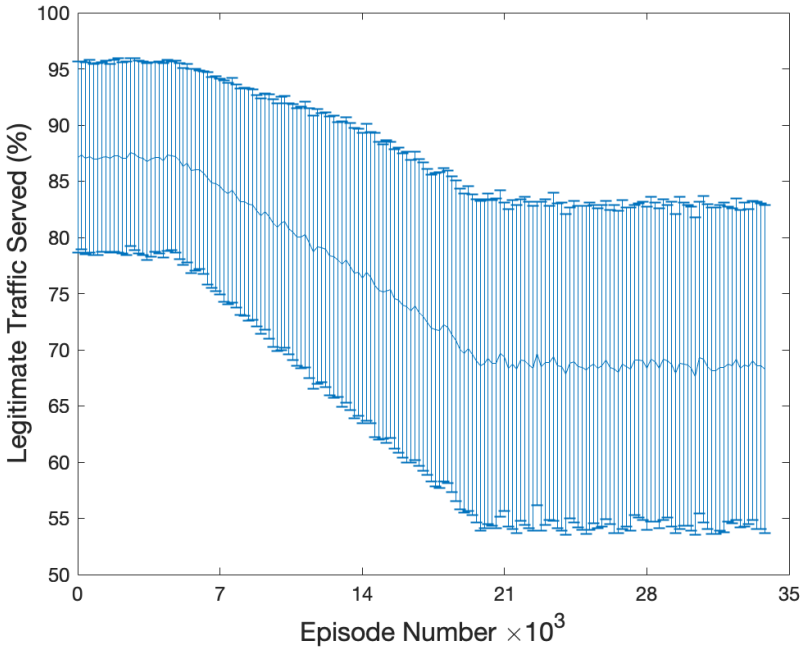


Figure 4.3: Percentage of traffic served during training of IDA against the Fair Throttle. Data is aggregated into bins of 1000 episodes where we display the mean and the standard deviation.

4.4.1 Evaluation Criteria

The focus of this chapter is the attacker, not the defender, our aim is to investigate how our intelligent attacks compare to the standard attacks. Chiefly, we are interested in the aims detailed in Sec 4.2, that is, in each evaluation where all attacks are provided the same resources, does IDA match or outperform all the standard attacks?

In each evaluation we use a separate adversary that has been trained against the defender to be evaluated. Evaluations display the mean of the percentage of legitimate traffic served and the variance of the mean over repetitions. The objective of this thesis is to either match or lower the mean percentage of traffic serviced of the most damaging standard attack for a given defender. Claims of significance in this thesis have be determined through a Paired Sample T-Test using all episodes from the 10 repetitions.

Like the analysis in Chapter 3 we provide both the mean and the variance of

the mean over 10 repetitions. Variance is caused through differing policies due to the randomly generated episodes during training for both MARL systems and IDA.

To demonstrate IDA's adaptability our experiment was repeated against a selection of the defenders explored in Chapter 3. Our primary focus is to demonstrate the IDA's capacity to adapt to differing defenders, however we do so in a way to provide analysis how the Fair Throttle and MARL perform against an intelligent attacker. We reintroduce a subset of the agents described in Sec 3.2.1 and Sec 3.2.3 for the purpose of evaluation. Further we introduce one new defender where we challenge IDA by allowing the defender to make four times the number of actions as the adversary. Our introduced defender is a variant of the previously described MARL-A system where the defender makes a move every 0.5 seconds however maintains the observation window of 2 seconds. The defender, MARL-S, represents a sliding window where the interval between actions and observation the window is detached addressing the vulnerability we identified in Sec 3.3.4. MARL-S adopts the proposed reward system outlined in Sec 3.2.2 as it was observed to increase performance against constant traffic in Sec 3.3.3. MARL-S does not incorporate the hierarchical structure due to the communication delays which would obstruct an immediate response.

Attacks Used

Each defender is subject to both likely and intelligent attacks. This chapter models intelligent attacks through IDA. To provide a comparison how IDA performs compared to typical attacks we reintroduce both the Constant Attack and the Short Burst attack from Sec 3.3.1. Attacks from the prior chapter that failed to outperform the effectiveness of the Short Burst or Constant Attack do not contribute to our analysis and are therefore omitted.

4.4.2 Evaluation results

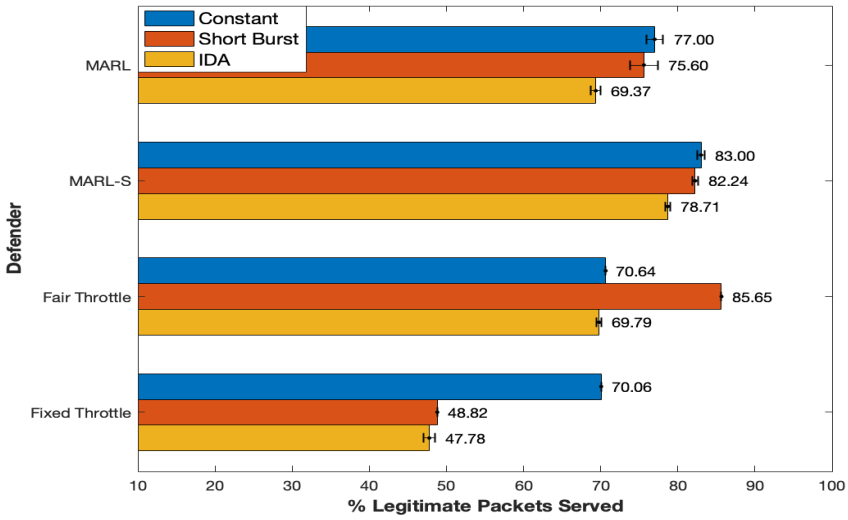
We now evaluate the rate-limiters in two different topologies consisting of 3 and 9 throttling agents. To limit the independent variables we use a single Handler in both topologies, the expansion of IDA is investigated in Sec 4.4.5. The hierarchical communication model is not designed for small topologies and is introduced in the 9 agent topology. Fig 4.4 shows the comparison of IDA against standard attacks for both agent topologies.

Name	Throttle Mechanism	Description	Investigative Reason
MARL	Drop Rate	The Reinforcement Learning rate-limiter by *Malialis and Kudenko*	We contribute to prior literature by extending analysis on MARL
MARL-DH	Drop Rate	MARL extension that uses DDQN to interpret a hierarchical state	In simulations with the bottleneck, MARL-DH outperformed MARL against all attacks outside the Short Burst
MARL-S	Drop Rate	MARL-A extension that generates an action every 0.5 seconds	We challenge IDA by presenting a defender that is able to react quicker than the adversary
Fair Throttle	Forwarding Rate	Centralised rate-limiter that uses the AIMD algorithm by *Yau et al.*	We contribute to prior literature by extending analysis on the Fair Throttle
Fixed Throttle	Drop Rate	A variant of the Fair Throttle that uses a drop rate like MARL as opposed to the forwarding rate	The Fixed Throttle is vulnerable to a Short Burst aligned with the monitoring window. We test the adversary's capacity to match or exceed this vulnerability.

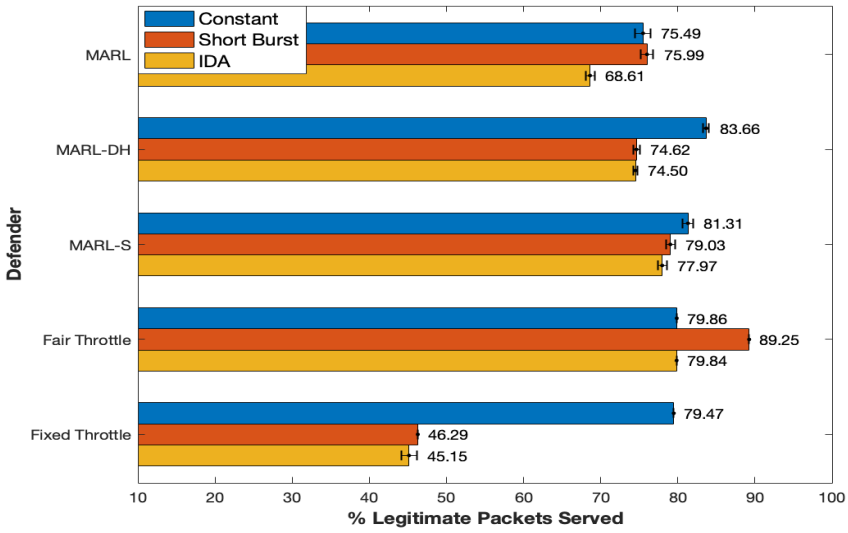
Table 4.1: Summary of different defenders to be evaluated by IDA. A Drop Rate assigns a percentage of traffic to be dropped until the following interval whilst a Forwarding Rate sets a maximum amount of traffic that can pass.

Attack Name	**Description**
Constant Attack	Attackers maintain constant traffic at the maximum rate
Short Burst	Attackers switch between maximum and zero capacity over a two second interval
IDA	Each host is delegated to one of the handlers of our Intelligent DDoS Adversary

Table 4.2: Evaluated Attacks for Chapter 4



(a) Comparison over 3 Agent Topology



(b) Comparison over 9 Agent Topology

Figure 4.4: Comparison of legitimate packets served by both IDA and standard attacks. IDA is seen to reduce the traffic served compared to all standard attacks.

Three Defender Topology

Against all defenders in the Three Defender Topology we observe IDA to outperform all standard attacks, these were results were statistically significant under a Paired Sample T-Test. It demonstrated a clear capacity to model an attack best suited for the defender given the provided resources. IDA achieves performance degradations against both the MARL (6.2%) and the MARL-S (3.5%) systems when compared to the most effective standard attack, indicating the existence of a more damaging attack pattern not represented by standard attacks. Despite the MARL-S having four times as many moves compared to IDA it identified an attack policy that exceeded common attacks. The differential between the traffic dropped between IDA and the most damaging constant attack is significant but limited for both the Fair and Fixed Throttle. This suggests that IDA was able to identify a policy that was outperformed the most damaging common attack, however a drop of in performance of less than 1% shows that the Fair Throttle is largely resistant to the intelligent attack.

Nine Defender Topology

The Nine Defender Topology presents a scalability challenge for both defenders and the adversary. This topology utilises 1 Handler, 9 Defenders, 18 Hosts and the server's lower and upper boundaries are increased to L_S=17 and U_S=20. The number of throttling agents increases the complexity of the task for the defender, MARL suffers scalability concerns due to the credit assignment problem [Malialis and Kudenko, 2013a], therefore we reintroduce MARL-DH which utilises a the hierarchical communication structure. Similarly this topology introduces a number of challenges for the attacker:

Challenges for the adversary:

- We have increased the number of Hosts up to 18, therefore the Handler must instruct a larger number of Hosts.
- There is a reduction in the Handler to Defender ratio, therefore the Handler must consider the possible moves of 9 different defenders.
- Both the increase of Hosts and Defenders has resulted in a large increase of the size of the state the Handler must interpret.

The evaluation reinforces observations from the three defender topology where IDA continued to outperform all attacks. This was statistically significant for all attacks outside the Fair Throttle which recorded a p-value of $p = 0.8$. IDA reduced the percentage of packets served by less than 0.15% for both MARL-DH and the Fair Throttle. By exhibiting a mean performance below the Constant Attack for the Fair Throttle we demonstrate there exists a subset of episodes where the Constant Attack is outperformed.

4.4.3 Analysing IDA

We now analyse the attacks produced by IDA to provide further understanding to researchers how an attacker may attack a system given available resources. Although this analysis is of benefit to understanding the Fair Throttle and MARL, the primary contribution is the methodology to evaluate future rate-limiters.

Attack Distributions

In Sec 4.4.2 we observed IDA outperform the Fair Throttle by a small margin. This contributes to current literature by demonstrating that the Fair Throttle is resistant to the intelligent attack, however it is unclear under what circumstances IDA outperforms the Constant Attack. Fig 4.5 compares the percentage of traffic serviced during an IDA attack against the Constant Attack for the same episode, episodes have been ordered by the percentage of traffic serviced under the Constant Attack. We observe IDA follows the Constant-Attack unless this would result in less than approximately 5% of legitimate traffic disrupted, in this case it would adopt a different attack pattern. A constant stream of DDoS traffic would be ineffective in high concentrations where the throttling agent is exposed to limited legitimate traffic. In these episodes by throttling the concentrated stream, the network can service legitimate traffic with minimal disruption. The divergence of packets served by IDA shows that for these episodes it determined it would be more effective to adopt a variable attack pattern to bypass the defence policy. This provides a clear demonstration of IDA's capacity to adapt its attack considering both the defender and available resources. As IDA outperforms the Constant Attack for only weak DDoS attacks we strengthen analysis that the Fair Throttle is resistant to the intelligent attacker.

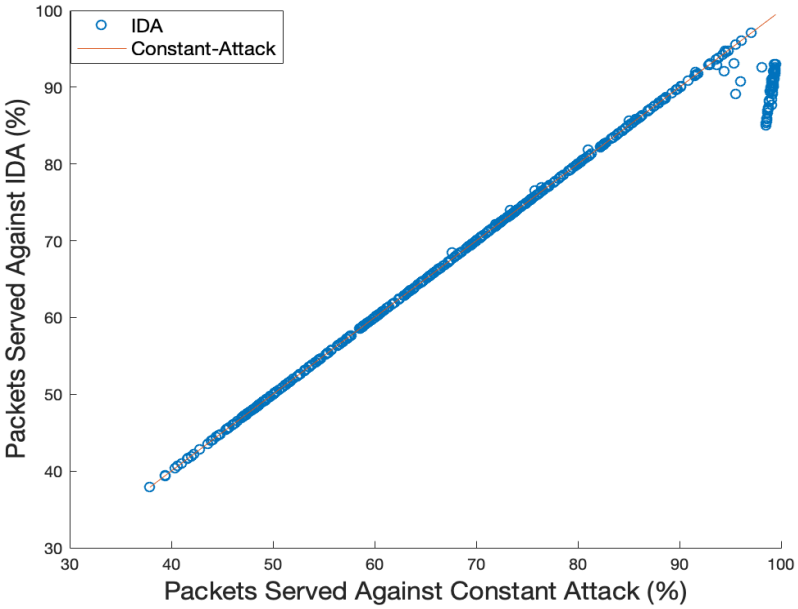
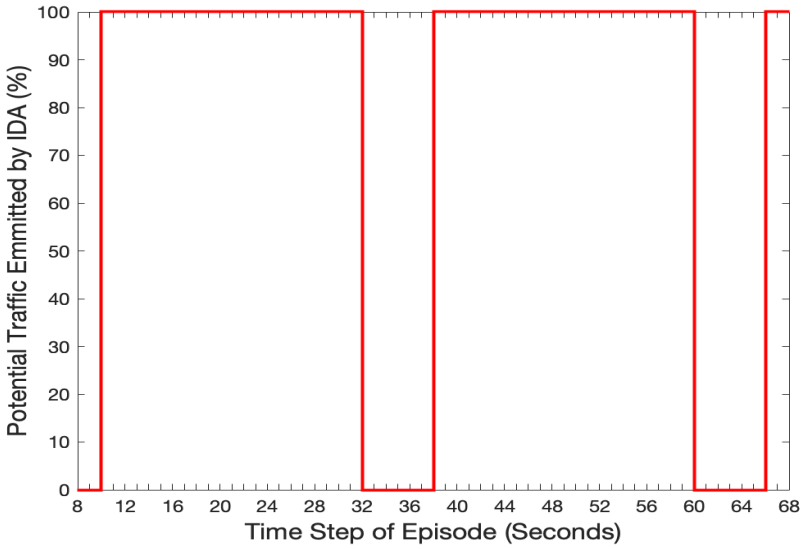


Figure 4.5: Divergence of network disruption between IDA and Constant Attack. IDA is seen to deviate when the Constant Attack would disrupt less than 5% of legitimate traffic.

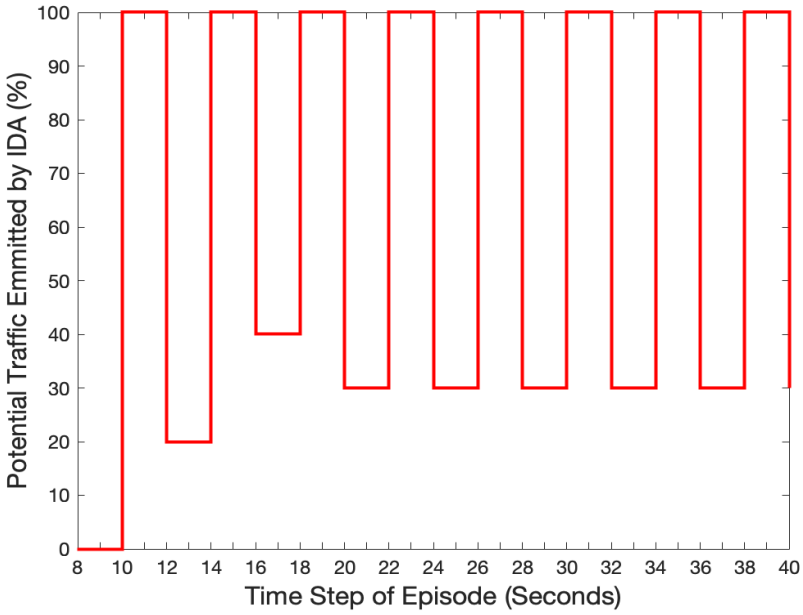
Analysing Attack Patterns

We seek to identify the exact attack patterns generated by the adversary to outperform the typical attacks. Fig 4.6a shows a snapshot of attack behaviour from an episode that outperformed the Constant Attack against the Fair Throttle. We observe that IDA performs a Constant Attack with 6 second intervals of no traffic. This would align with the AIMD algorithm deactivating the throttle after two consecutive intervals below the server's lower-boundary where there has been no large change in traffic. Here it is clear that IDA has learnt a policy that exploits the functionality of the AIMD algorithm, which directs the Fair Throttle, and chose to sacrifice 6 seconds of no disruption to reset the throttle. The AIMD algorithm must converge to the appropriate throttling rate, here we establish a potential area of improvement for the Fair Throttle where a weak attack can exploit the limited throttle memory.

The snapshot against MARL in Fig: 4.6b shows a unique variant of the



(a) Observed Long Burst Variant generated by IDA against the Fair Throttle



(b) IDA mimics legitimate traffic loads to bypass MARL

Figure 4.6: Observed attack patterns by IDA against the Fair Throttle and MARL

Short-Pulse converging on a lower bound of 30% during the 'off-period'. An emission of 30% of the adversarial potential would see similar rates produced as seen by legitimate traffic; this suggests the adversary found mimicking user traffic loads to be advantageous against MARL. Both of these attack patterns deviate from standard attacks patterns and are examples of IDA's capacity to identify detrimental vulnerabilities through exploration.

The analysis of attack patterns can aid defence design by alerting the researcher to invalid assumptions or unseen vulnerabilities not evident through standard evaluation. Whilst IDA can be used as a comparative evaluation tool we argue such analysis would benefit the design of resistant defenders. We stress that the displayed behaviour is not reflective of the entire attack, we have established IDA to vary its attack based on available resources.

4.4.4 Isolating Defence Mechanisms

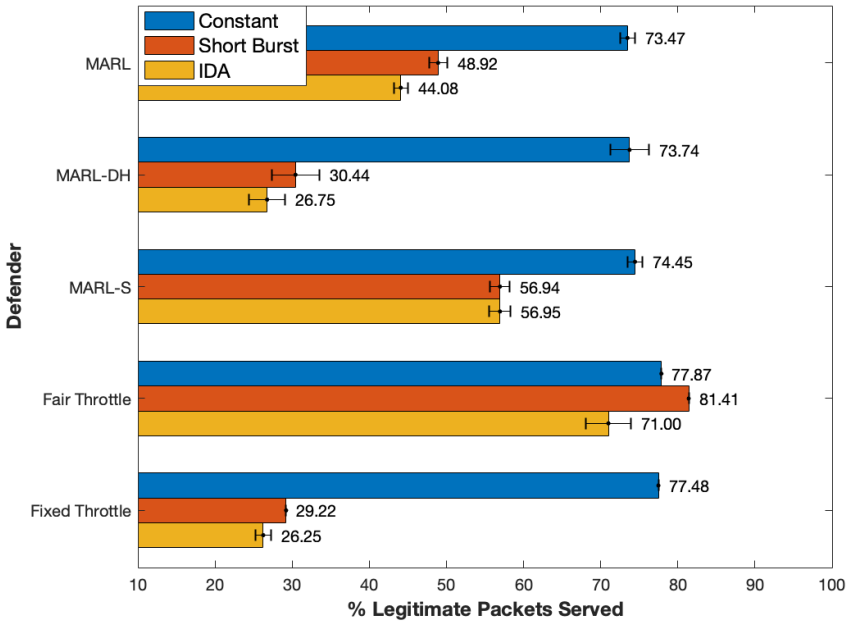


Figure 4.7: IDA evaluates the capacity of the defenders to avoid congestion on the 9 agent topology

In Sec 3.3.4 we evaluated the utility of individual defences through the re-

Number of Handlers	1 (IDA)	2	3	4	5
FairThrottle	79.84	79.96	79.92	80.17	80.69
FixedThrottle	45.15	45.92	46.27	45.34	46.33
MARL	68.61	69.85	70.44	71.32	71.41
MARL-S	77.97	77.99	77.18	77.55	77.62
MARL-DH	74.69	74.69	74.98	75.09	75.14

Table 4.3: Percentage of legitimate traffic served as the number of Handlers is increased. We observe for IDA variants that employ additional Handlers, the effectiveness of the attack either stagnates or decreases.

moval of the bottleneck. We perform a similar experiment with the goal of investigating IDA's capacity to adapt to differing goals. Reinforcement Learning is 'model free', this enables the adversary to adapt to differing goals by replacing the reward mechanism. By failing to service any packets during congestion IDA is assessing whether the proposed defenders can be bypassed and instead relying on secondary defences. In other words, IDA is determining the utility of an individual defence mechanism to the system.

Fig 4.7 evaluates the rate-limiting defenders in the 9 defender topology where one second intervals of congestion result in no packets served. To reflect the goal of avoiding server congestion, MARL-DH and MARL-S use the original reward system where the defender receives a reward of -1 when they fail to avoid congestion. IDA outperforms standard attacks against all rate-limiters outside MARL-S, the Short Burst outperformed IDA for MARL-S at $p = 0.54$ which is not statistically significant. The Fair Throttle, which has been resistant to variable attacks, demonstrates a 3.5% decrease in performance than the Constant Attack providing further information to researchers on the versatility of the defenders. The adaptability of IDA would allow the researcher to perform similar versatility analysis with differing performance outcomes for congestion.

4.4.5 Increasing Handlers

We have so far limited the number of Handlers during an attack. Additional Handlers would provide the adversary with increased ability to commit to multiple traffic patterns if it were advantageous for the adversary. Our experiments have been limited due to the computational expense from multiple Deep Learning agents as well as the extensive network simulation needed for training would require a distributed framework. Although we have designed our system to allow

multiple Handlers there remains the question whether allowing more individualised attack patterns provides any benefit to the adversary.

In this experiment we assess whether our current design benefits from the addition of more Handlers. We retain the 9 defender topology and use an IDA variant where we increase the number of Handlers. Table 4.3 displays the percentage of legal traffic serviced for the 9 agent topology as we increase the number of Handlers from 1 to 5. We observe that the performance of the attack either stagnates or decreases as more Handlers are added. Therefore we identify that the current design of the adversary incurs a scalability challenge with the introduction of more Handlers.

4.5 Discussion and Conclusion

In this chapter we introduced our novel Intelligent DDoS Adversary, which uses Deep Reinforcement Learning to learn attack policies designed to optimise congestion. Our research provides a solution to the systematic omission of current evaluation practices which fails to consider the intelligent attacker. Our design reflects the DDoS design detailed by *Lin et al.* [Lin and Tseng, 2004] to provide an accurate representation of how an adversary may attack a network. Through learning and evaluating via network simulation, our adversary is compatible with standard practice evaluation techniques. We consider IDA to be a component of the arsenal of evaluation tools for comparing a system, an administrator would therefore need to consider both the performance against the intelligent and expected attacks.

We then contributed to current literature by analysing the performance and versatility of a series of rate-limiting mechanisms against intelligent attacks. Prior evaluations of the MARL and Fair-Throttle have neglected the intelligent attacker and therefore our analysis provides information relating to the utility of these two systems. Against all rate-limiting defenders and throughout all tested topologies, IDA either matched or outperformed the most damaging standard attack when given the same set of episodes. It demonstrated a clear ability to consider both the functionality of the defender and the provided resources in its decision making. Our investigations showed that whilst variants of the MARL system can outperform the Fair Throttle against constant traffic, it demonstrated a significant vulnerability to an intelligent attack. Conversely it strengthened the analysis that the Fair-Throttle is resistant to variable traffic

loads.

IDA was then used to provide insights on the cause of the exposed vulnerabilities though examination of IDA's attack behaviour. We demonstrated by exploring when a variable attack was advantageous against the Fair Throttle and the cause for the exposed vulnerability of MARL. In regards to the Fair-Throttle our experiments show that a constant attack represented a lower bound on expected performance with exception to concentrated attacks where a Constant Attack would be ineffective. We then demonstrated IDA's exploitation of MARL's learnt associations of expected traffic loads for both legitimate traffic and the adversary. Through exploration IDA discovered by mimicking legitimate user traffic during the off period of a burst attack it could deactivate the throttle for the subsequent interval. The investigation of the traffic behaviour of IDA is of benefit to future researchers that seek to address the exposed vulnerability of MARL. The methodology detailed aids future researchers to design resistant rate-limiting defenders.

Lastly, we investigated expansions to IDA such as evaluating the individual utility of a mechanism. Defence mechanisms do not operate in isolation and therefore poor performance in similar investigations do not infer the model is ineffective. Such analysis however does provide the researcher with understanding on whether particular mechanisms can be bypassed to help us identify the versatility of the system. By investigating performance with additional Handlers we identified a scalability challenge for our current approach. We chose more Handlers to enable a decentralised structure that address dimensionality concerns for either larger states or to allow simultaneous attack patterns. The lack of direct communication between Handlers allows for all actions to be generated simultaneously allowing for a decentralised design. Whilst the approach mitigated one challenge it incurred slight degradation of performance. We address potential research avenues to address scalability concerns in Sec 5.2.3.

Chapter 5

Conclusion and Future Work

5.1 Overview

The increasing sophistication of the DDoS Attacker and shifting nature of the internet has motivated us to investigate the role of Reinforcement Learning in creating DDoS defences resistant to developments of traffic behaviour. This thesis explored developing an adaptive defender though examining the MARL Router Throttling system and introduced a novel automated evaluation metric that approximates the effectiveness of the defender against an intelligent attacker.

In Chapter 2 we explored prior proposals that incorporated Reinforcement Learning to combat the DDoS threat. We identified that current approaches have not demonstrated sufficient performance in a real network or lack a reliable reward calculator to allow the defender to continue adapt from emulation into the deployed network. Our review focussed on MARL, a Reinforcement Learning rate-limiter where we identified the need for further investigation to establish whether it is a satisfactory alternative to the Fair Throttle. In observing that current automated evaluation practices do not consider the intelligent attacker we identified that a novel use of Reinforcement Learning would be to evaluate proposed DDoS defences.

Network defence continues to require frequent updates as attackers become more sophisticated and are often reactive to developments in attack behaviour.

Current evaluation methodology simulates observed attack behaviour against the defence mechanism based on the expectation how the adversary will behave. The approach provides comparative analysis of defence mechanisms however omits critical information such as how the defender performs against the intelligent attacker. To limit unforeseen vulnerabilities, researchers must consider how the intelligent attack will behave, current approaches such as individual analysis of a defender are costly and not comparative. In this work we presented an automated approach to approximating the effectiveness and resilience against the Intelligent DDoS Adversary through Reinforcement Learning. Our simulated adversary considered the functionality of the defender and availability of resources to generate attack patterns designed to cause congestion. This thesis demonstrated the utility of the adversary through the lenses of a comparative review between the MARL and the Fair Throttle rate-limiting systems.

Our intelligent adversary is a novel evaluation tool that exceeds prior evaluation practices by providing a metric that approximates how an intelligent attacker will behave. The adversary is compatible with prior evaluation practices as it uses network simulation to train and evaluate a DDoS defender. We enable evaluations where the availability of resources is static allowing comparative evaluations where the adversary is restricted to the same resources against all defenders.

We demonstrated one such version of the adversary, which we named IDA, against the MARL and Fair Throttle rate limiting systems. Our analysis considered variants of the replicated models which included proposed improvements of the MARL system. Against all the rate limiters we tested and throughout all topologies evaluated, IDA was demonstrated to use the same DDoS resources to generate attack patterns that either matched or exceeded the network degradation caused by standard attacks. By matching or outperforming the most damaging typical attack we contribute to evaluation practices by providing a tool that approximates the worst case performance mitigating the need for the researcher to evaluate the defender under all possible attack patterns.

Our investigation extends the prior understanding of the effectiveness of the MARL and Fair Throttle rate limiting systems. We contradicted the prior comparative evaluation provided by *Malialis and Kudenko* by demonstrating the Fair Throttle significantly outperforms MARL against Short Burst attacks. Using standard evaluation practices we identified MARL's reliance on the existence of a bottleneck to manage congestion against variable traffic thus identifying a

versatility challenge for MARL. By acknowledging MARL's reliance of the bottleneck we provided two alternative systems, which considered Deep Reinforcement Learning and encouraged cooperation with tandem defence mechanisms, these were demonstrated to increase performance against constant traffic attacks however were also vulnerable to the Short Burst. By extending the comparison by introducing the intelligent adversary we identified a new vulnerability in the MARL defender, conversely the Fair Throttle was shown to be resistant to the intelligent attack. Therefore we conclude that the current design of MARL is in insufficient to be considered a viable alternative to the Fair Throttle.

We contribute to the design of resilient DDoS defenders by illustrating how the adversary can be used to identify the causes of the exposed vulnerabilities. We did so by analysing the attack patterns generated by IDA where we identified a unique Burst Attack that was able to bypass MARL. This is of particular use to future research of MARL as it demonstrated that the solitary state-space could be exploited. Similar analysis provided insights for the Fair Throttle regarding to the distribution of attack resources where a Burst Attack outperformed the Constant Attack. The contribution we provide exceeds the value of the individual analysis of the replicated systems. Our work establishes methodology where a model free adversary is used to identify and illustrate defence vulnerabilities for future rate-limiters. By providing a tool that enables the researcher to understand how the adversary was able to exploit the defence mechanism, we assist the design of future rate-limiters resistant to the intelligent attacker.

In summary we have proposed a novel use of Reinforcement Learning which enables the evaluation of a rate-limiting defender against an intelligent adversary. The established methodology can be used to assist the design of rate-limiters resistant to the intelligent adversary. The adversary was shown to either match or outperform standard attacks against all rate-limiters in all topologies. Analysis of the attacks generated can provide insights for the causes of the identified vulnerabilities. Our implementation of the adversary was used to evaluate the MARL and Fair Throttle rate-limiting systems. We demonstrated that the Fair Throttle is resistant to the intelligent attacker unlike MARL and concluded that MARL is an unsuitable alternative to the Fair Throttle.

5.2 Future Work

5.2.1 Designing Adaptive Rate-Limiters

Our investigation concluded that MARL is not a suitable replacement to the Fair Throttle as it was unable to counter variable traffic. We identified that the limited state-space and the use of a fixed drop rate as opposed to a forwarding rate contributed to this vulnerability. Our DDQN adaptation, MARL-DH, outperformed the Fair Throttle against attacks with large sections of constant traffic demonstrating a Reinforcement Learning implementation can outperform the Fair Throttle. Future research would benefit from adopting the structural advantageous of the Fair Throttle such as a forwarding rate that adapts to changes in traffic and the extension of the state-space to consider the variance of traffic.

5.2.2 Adapting for a Continuous Action Space

Our adversary was limited through the discretisation of a continuous action space into 11 actions limiting the breadth of actions available to each Handler. The MARL defender was similarly restricted in the availability of throttling actions. The functionality of either system could be enhanced through the use of a learning agent that provides a continuous action space. *Gupta et al.* have demonstrated a viable Deep Reinforcement Learning actor-critic agent that allows a continuous action space in a multi-agent environment [Gupta et al., 2017].

5.2.3 Addressing Scalability Concerns

In Sec 4.4.5 we observed that expansions of IDA, where the number of Handlers increased, suffered scalability concerns where performance stagnated or dropped. Our architecture reflects the typical DDoS attacks where attacking Hosts are delegated to Handlers to allow a distributed attack over a vast geographical region. The decentralised approach also happened to reduce the exponential growth in actions that are caused by allowing individualised attack patterns from each Handler. We have identified the incompatibility of experience replay with simultaneous learners and the use of a global reward as causes for the observed scalability challenge. Deep Reinforcement Learning samples past experiences through Experience Replay to avoid unreliable and slow learning

[Riedmiller, 2005]. Experience Replay assumes a constant environment which is challenged by the existence of multiple agents simultaneously learning [Foerster et al., 2017]. The sampling of past rewards would see the agent repeatedly rewarded for scenarios not reflective of the current environment. *Foerester et al.* have provided two possible approaches to mitigate disadvantageous sampling that may be on benefit to larger simulations [Foerster et al., 2017]. Larger simulations are likely to suffer challenges from the use of a global reward. A global reward for a small number of agents encourages cooperation however larger systems may experience the credit-assignment problem, comparative mechanisms that balance individual rewards and cooperative behaviour are well documented [Panait and Luke, 2005].

An alternative approach would be to avoid multiple learning agents by centralising decision making into fewer Handlers. This would result in the learning agent being responsible for the actions of multiple Handlers requiring more training and incurring a new challenge referred to as the 'curse of dimensionality' where a single learning agent would be responsible for interpreting the entire state and all actions. There has been limited research in Reinforcement Learning for large discrete action spaces which may pose a challenge for more sophisticated environments, *Dulac-Arnold et al.* demonstrated an Actor-Critic policy architecture where an agent with up to 1 million actions was able to formulate an effective policy [Dulac-Arnold et al., 2015].

5.2.4 Application for a General Network Evaluator

Our investigation was limited to evaluating rate-limiting systems where the adversary considered the volume of traffic produced. The challenge is to expand the adversary to evaluate defences that consider the contents of packets. As Reinforcement Learning becomes more sophisticated the capacity of simulating an expanded adversary, not limited to traffic levels, becomes feasible. The model free learning system can easily be adapted for similar tasks requiring modifications to the reward mechanism to reflect the new goal. Therefore there exists the potential to use Reinforcement Learning as a blackbox penetration testing tool for network defence.

5.2.5 Incorporation into Standardised Test Beds

Our adversary represents a step in the direction of establishing realistic evaluation by considering the intelligent attacker. A possible criticism of our ex-

periment is that our network simulation did not emulate a realistic network. The establishment of a standard practice evaluation methodology, reducing the discrepancy between emulation and the real world, remains an ongoing pursuit [Mirkovic et al., 2009]. A natural direction for future research is to incorporate our adversary into established evaluation benchmarks like the DETER test bed [Mirkovic et al., 2007] where the model free design of Reinforcement Learning would allow the adversary to adapt to future DDoS defenders. The incorporation of the adversary into benchmark test beds would provide a realistic adversary and increase the utility of such test beds for researchers by allowing them to simulate against an intelligent attacker.

5.2.6 Threat of an Online Adversary

The use of the adversary in a closed environment solved the challenge of the calculation of accurate rewards. By designing a reliable reward calculator that measures network disruption, our adversary could be adapted to coordinate an actual DDoS attack. The cost of an unsuccessful DDoS attack for the attacker is modest when compared to the cost of a successful attack to the victim, therefore an attacker is more likely to afford the lengthy training time required for online learning rather than the defender. An adversary could be used for online penetration testing, or used by attackers to produce sophisticated DDoS attacks. It would be of interest for researchers to quantity the feasibility of an attacker employing Reinforcement Learning in order to understand a potential future DDoS threat.

Bibliography

[Buşoniu et al., 2010] Buşoniu, L., Babuška, R., and De Schutter, B. (2010). Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications*, pages 183–221. Springer.

[Cannady, 1998] Cannady, J. (1998). Artificial neural networks for misuse detection. In *National information systems security conference*, volume 26. Baltimore.

[Cisco, 2017] Cisco (2017). Comparing traffic policing and traffic shaping for bandwidth limiting. `https://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-policing/19645-policevsshape.html`. Accessed: 2019-05-04.

[Dulac-Arnold et al., 2015] Dulac-Arnold, G., Evans, R., van Hasselt, H., Sunehag, P., Lillicrap, T., Hunt, J., Mann, T., Weber, T., Degris, T., and Coppin, B. (2015). Deep reinforcement learning in large discrete action spaces. `https://arxiv.org/pdf/1512.07679`.

[Fayaz et al., 2015] Fayaz, S. K., Tobioka, Y., Sekar, V., and Bailey, M. (2015). Bohatei: Flexible and elastic ddos defense. In *24th Security Symposium*, volume 24, pages 817–832.

[Foerster et al., 2017] Foerster, J., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H., Kohli, P., and Whiteson, S. (2017). Stabilising experience replay for deep multi-agent reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1146–1155. JMLR. org.

[Gligor, 1984] Gligor, V. D. (1984). A note on denial-of-service in operating systems. *IEEE Transactions on Software Engineering*, 3:320–324.

[Gupta et al., 2017] Gupta, J. K., Egorov, M., and Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 66–83. Springer.

[Hadfield-Menell et al., 2017] Hadfield-Menell, D., Milli, S., Abbeel, P., Russell, S. J., and Dragan, A. (2017). Inverse reward design. In *Advances in neural information processing systems*, volume 31, pages 765–774.

[Han et al., 2017] Han, G., Xiao, L., and Poor, H. V. (2017). Two-dimensional anti-jamming communication based on deep reinforcement learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2087–2091. IEEE.

[Javaid et al., 2016] Javaid, A., Niyaz, Q., Sun, W., and Alam, M. (2016). A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies*, volume 9, pages 21–26. ICST.

[Kotenko and Ulanov, 2014] Kotenko, I. and Ulanov, A. (2014). Agent-based simulation of ddos attacks and defense mechanisms. *International Journal of Computing*, 4:113–123.

[Li et al., 2017] Li, Y., Quevedo, D. E., Dey, S., and Shi, L. (2017). Sinr-based dos attack on remote state estimation: A game-theoretic approach. *IEEE Transactions on Control of Network Systems*, 4:632–642.

[Lin and Tseng, 2004] Lin, S.-C. and Tseng, S.-S. (2004). Constructing detection knowledge for ddos intrusion tolerance. *Expert Systems with applications*, 27:379–390.

[Mahajan et al., 2002] Mahajan, R., Bellovin, S. M., Floyd, S., Ioannidis, J., Paxson, V., and Shenker, S. (2002). Controlling high bandwidth aggregates in the network. *ACM SIGCOMM Computer Communication Review*, 32:62–73.

[Malialis et al., 2015] Malialis, K., Devlin, S., and Kudenko, D. (2015). Distributed reinforcement learning for adaptive and robust network intrusion response. *Connection Science*, 27:234–252.

[Malialis and Kudenko, 2013a] Malialis, K. and Kudenko, D. (2013a). Large-scale ddos response using cooperative reinforcement learning. In *11th European Workshop on Multi-Agent Systems (EUMAS)*, volume 11, pages 1–14. European Workshop on Multi-Agent Systems.

[Malialis and Kudenko, 2013b] Malialis, K. and Kudenko, D. (2013b). Multiagent router throttling: Decentralized coordinated response against ddos attacks. In *Twenty-Fifth IAAI Conference*, volume 25, pages 1551–1556. Innovative Applications of Artificial Intelligence.

[Matthews, 2014] Matthews, T. (2014). What ddos attacks really cost businesses? Technical report, Incapsula.

[Meade et al., 2017] Meade, B., Lafayette, L., Sauter, G., and Tosello, D. (2017). Spartan hpc-cloud hybrid— delivering performance and flexibility. `https://doi.org/10.4225/49/58ead90dceaaa`.

[Mirkovic et al., 2006] Mirkovic, J., Arikan, E., Wei, S., Thomas, R., Fahmy, S., and Reiher, P. (2006). Benchmarks for ddos defense evaluation. In *MILCOM 2006-2006 IEEE Military Communications conference*, pages 1–10. IEEE.

[Mirkovic et al., 2009] Mirkovic, J., Fahmy, S., Reiher, P., and Thomas, R. K. (2009). How to test dos defenses. In *2009 Cybersecurity Applications & Technology Conference for Homeland Security*, pages 103–117. IEEE.

[Mirkovic and Reiher, 2004] Mirkovic, J. and Reiher, P. (2004). A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34:39–53.

[Mirkovic et al., 2007] Mirkovic, J., Wei, S., Hussain, A., Wilson, B., Thomas, R., Schwab, S., Fahmy, S., Chertov, R., and Reiher, P. (2007). Ddos benchmarks and experimenter's workbench for the deter testbed. In *2007 3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities*, pages 1–7. IEEE.

[Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518:529–531.

[Needham, 1994] Needham, R. M. (1994). Denial of service: an example. *Communications of the ACM*, 37:42–46.

[NetScout, 2018] NetScout (2018). Cloud in the crosshairs — netscout's 14th annual worldwide infrastructure security report. `https://www.netscout.com/report`. Accessed: 2019-05-03.

[Nguyen and Choi, 2010] Nguyen, H.-V. and Choi, Y. (2010). Proactive detection of ddos attacks utilizing k-nn classifier in an anti-ddos framework. *International Journal of Electrical, Computer, and Systems Engineering*, 4:247–252.

[Panait and Luke, 2005] Panait, L. and Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems*, 11:387–434.

[Peng et al., 2007] Peng, T., Leckie, C., and Ramamohanarao, K. (2007). Survey of network-based defense mechanisms countering the dos and ddos problems. *ACM Computing Surveys (CSUR)*, 39:1–42.

[Riedmiller, 2005] Riedmiller, M. (2005). Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pages 317–328. Springer.

[Servin and Kudenko, 2008] Servin, A. and Kudenko, D. (2008). Multi-agent reinforcement learning for intrusion detection: A case study and evaluation. In *German Conference on Multiagent System Technologies*, pages 97–103. Springer.

[Shiva et al., 2010] Shiva, S., Roy, S., and Dasgupta, D. (2010). Game theory for cyber security. In *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, volume 6, pages 34–38. ACM.

[Sutton, 2019] Sutton, R. S. (2019). Tile coding software – reference manual, version 3 beta. `http://incompleteideas.net/tiles/tiles3.html`. Accessed: 2018-09-13.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Introduction to reinforcement learning*. MIT press.

[Van Hasselt et al., 2016] Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, volume 30, pages 2094–2100.

[Vissers et al., 2015] Vissers, T., Van Goethem, T., Joosen, W., and Nikiforakis, N. (2015). Maneuvering around clouds: Bypassing cloud-based security providers. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1530–1541. ACM.

[Wang et al., 2017] Wang, A., Mohaisen, A., and Chen, S. (2017). An adversary-centric behavior modeling of ddos attacks. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 1126–1136. IEEE.

[Wu et al., 2010] Wu, Q., Shiva, S., Roy, S., Ellis, C., and Datla, V. (2010). On modeling and simulation of game theory-based defense mechanisms against dos and ddos attacks. In *Proceedings of the 2010 spring simulation multiconference*, pages 159–167. Society for Computer Simulation International.

[Xu et al., 2007] Xu, X., Sun, Y., and Huang, Z. (2007). Defending ddos attacks using hidden markov models and cooperative reinforcement learning. In *Pacific-Asia Workshop on Intelligence and Security Informatics*, pages 196–207. Springer.

[Yan et al., 2012] Yan, G., Lee, R., Kent, A., and Wolpert, D. (2012). Towards a bayesian network game framework for evaluating ddos attacks and defense. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 553–566. ACM.

[Yang et al., 2005] Yang, X., Wetherall, D., and Anderson, T. (2005). A dos-limiting network architecture. In *ACM SIGCOMM Computer Communication Review*, volume 35, pages 241–252. ACM.

[Yau et al., 2005] Yau, D. K., Lui, J. C., Liang, F., and Yam, Y. (2005). Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. *IEEE/ACM Transactions on Networking*, 13(1):29–42.

[Zhang et al., 2015] Zhang, L., Zhang, H., Li, C., and Ni, B. (2015). Optimal jamming attack scheduling in networked sensing and control systems. *International Journal of Distributed Sensor Networks*, 11:31–39.