




Article

AALLA: Attack-Aware Logical Link Assignment Cost-Minimization Model for Protecting Software-Defined Networks against DDoS Attacks

Sameer Ali ^{1,2,*}, Saw Chin Tan ¹, Ching Kwang Lee ³, Zulfadzli Yusoff ³ , Muhammad Reazul Haque ¹, Alexios Mylonas ⁴  and Nikolaos Pitropakis ⁵ 

¹ Faculty of Computing & Informatics (FCI), Multimedia University (MMU), Cyberjaya 63100, Malaysia; sctan1@mmu.edu.my (S.C.T.); reazul@ieee.org (M.R.H.)

² Department of Information Technology, SZABIST University, Karachi 75600, Pakistan

³ Faculty of Engineering (FOE), Multimedia University (MMU), Cyberjaya 63100, Malaysia; cklee@mmu.edu.my (C.K.L.); zulfadzli.yusoff@mmu.edu.my (Z.Y.)

⁴ School of Physics, Engineering and Computer Science (SPECS), University of Hertfordshire, Hatfield AL10 9AB, UK; a.mylonas@herts.ac.uk

⁵ School of Computing, Engineering & the Build Environment, Edinburgh Napier University, Edinburgh EH10 5DT, UK; n.pitropakis@napier.ac.uk

* Correspondence: sameer.ali@szabist.edu.pk

Abstract: Software-Defined Networking (SDN), which is used in Industrial Internet of Things, uses a controller as its “network brain” located at the control plane. This uniquely distinguishes it from the traditional networking paradigms because it provides a global view of the entire network. In SDN, the controller can become a single point of failure, which may cause the whole network service to be compromised. Also, data packet transmission between controllers and switches could be impaired by natural disasters, causing hardware malfunctioning or Distributed Denial of Service (DDoS) attacks. Thus, SDN controllers are vulnerable to both hardware and software failures. To overcome this single point of failure in SDN, this paper proposes an attack-aware logical link assignment (AALLA) mathematical model with the ultimate aim of restoring the SDN network by using logical link assignment from switches to the cluster (backup) controllers. We formulate the AALLA model in integer linear programming (ILP), which restores the disrupted SDN network availability by assigning the logical links to the cluster (backup) controllers. More precisely, given a set of switches that are managed by the controller(s), this model simultaneously determines the optimal cost for controllers, links, and switches.

Keywords: internet of things; distributed denial of service; software-defined networks; controller; ILP; AALLA



Citation: Ali, S.; Tan, S.C.; Lee, C.K.; Yusoff, Z.; Haque, M.R.; Mylonas, A.; Pitropakis, N. AALLA: Attack-Aware Logical Link Assignment Cost-Minimization Model for Protecting Software-Defined Networks against DDoS Attacks. *Sensors* **2023**, *23*, 8922. <https://doi.org/10.3390/s23218922>

Academic Editors: Nancy Alonistioti, Spyros Panagiotakis and Evangelos K. Markakis

Received: 17 August 2023

Revised: 25 October 2023

Accepted: 26 October 2023

Published: 2 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Software-Defined Networking (SDN) has been attracting attention in data centre network operators, academia, and industry for its programmability and agility. SDN empowers smart industries, such as Industrial Internet of Things, with central network device configuration and administration by providing a global view of the network. The SDN framework is regarded as the hardware-less networking paradigm in which networking through programming is possible. Compared to traditional networking, in SDN technology the control and data planes are decoupled, which make it more agile in terms of networking management. A controller is responsible for the management of the entire network, whereas networking switches are responsible for operating based on the instructions deployed through controllers [1]. One of the factors for network performance and scalability is how the network is being designed [2]. The SDN architecture is flexible and can be programmed using any high-level programming language to serve the purpose of the

client devices and end-users [3]. The SDN platform is not only capable of providing high performance, but also providing energy efficiency and network security [4]. However, it is necessary to counter Distributed Denial of Service (DDoS) attacks by employing controller clustering methods to control the efficiency and performance from the view of entire network security [3,4]. Due to the central location of the controller, many security concerns caused by a single point of failure have been reported [5]. Firstly, the SDN control plane is unable to handle all the flow requests due to resource consumption or malicious traffic resulting from DDoS. Secondly, the fake flow request from switches can generate several unnecessary flow rules, which makes it difficult for the data plane to store flow rules for a normal flow of traffic [6,7].

In this research study, the logical links in AALLA model are used to provide connectivity and restore the availability of resources when DDoS attacks happen. The AALLA model considers a link assignment technique and is capable of restoring the network service availability under a disruption of existing links. When a given switch is affected by a DDoS attack, the logical links will take up the switch using the backup links connecting another available port on the switch. This will restore or resume the disrupted service again to the requested users, ensuring service availability. The past literature has focused on the security of controller placement, the security of message transformation, bandwidth optimization, and network scalability [8–19]. However, the past literature has not focused on link assignment strategies considering bandwidth and cost optimization under single points of failure in SDN networks. AALLA considers metrics such as latency, throughput, cost optimization for links, switches, and controllers along with the high availability (HA) of network services in the SDN environment.

Security has been regarded as a detrimental factor in the development of SDN networks [20]. Among the security requirements of SDN networks, uninterrupted availability is critical since the core function of SDN is to provide uninterrupted network services and resources. DDoS flooding attacks are the culprit in destroying availability in SDN networks [20,21]. DDoS attacks are created by two or more systems or botnets. A botnet is a compromised host system created when a computer is penetrated by software from a malware code [20]. It is essential to ensure SDN network availability for its end-users under DDoS flooding attacks. Current DDoS attacks have many forms, e.g., consumption of computational resources, disruption of configuration information, etc. [22]. To improve scalability and performance and avoid a single point of failure, the control plane is implemented as a distributed system with a cluster of controllers [23]. The hierarchy of controllers using controller clustering system is proposed as shown in Figure 1. More than one controller in SDN will serve as backup support controllers and also distribute the load of flow requests from switches.

The controller cluster is an SDN failover mechanism and proposes attack-aware logical link assignment from switches to the cluster (backup) controller under DDoS attacks. Our contributions are to formulate the logical link assignment using integer linear programming (ILP) with the intention of minimizing the cost of controllers, switches, and links. The derived model will provide a necessary tool to restore the SDN network to overcome a single point of failure.

This research paper makes the following contributions:

- We introduce the AALLA model, which aims to address the single-point-of-failure susceptibility in SDN networks exploited by DDoS attacks. The model utilizes logical link assignment from switches to backup controllers to restore the network's availability.
- We formulate the AALLA model as an integer linear programming (ILP) problem. The model simultaneously determines the optimal cost for controllers, links, and switches while restoring the disrupted SDN network.
- Our model specifically aims to minimize the cost of controllers, links, and switches in the SDN network.

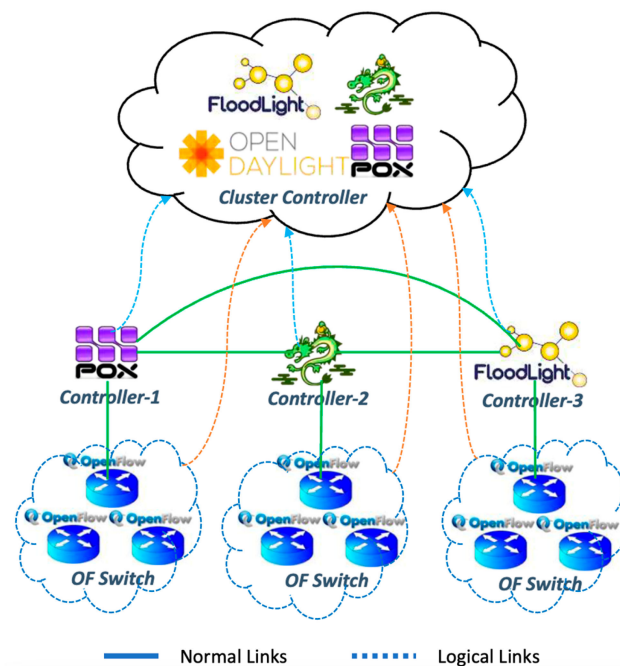


Figure 1. Controller cluster using logical link assignment in SDN network.

By formulating the problem as an ILP, we provide a tool that can be used to optimize the allocation of resources to the requested end-users in order to overcome the single point of failure and ensure availability of services. The rest of this paper is organized as follows. Section 2 reviews the related work on the SDN and DDoS attacks. The proposed attack-aware logical link assignment (AALLA) model is presented in Section 3 followed by the simulation results in Section 4. Conclusions are given in Section 5.

2. Related Work

In this section, background work on SDN, cluster controllers, and DDoS attacks and their security-related problems and possible solutions are discussed.

The authors in [24] present a heuristic approach Pareto-based Optimal Controller (POCO), a mathematical model for small- and medium-sized networks that provides operators with Pareto optimal placements with respect to different performance metrics. The authors in [25–27] present an integer linear model that proposes Survivor, a controller placement strategy that considers path diversity, capacity, and failover mechanisms in network design. In [28,29], the authors proposed a mathematical model for the controller placement in SDN that determines the optimal number, location, and type of controllers. The goal of the model is to minimize the cost of the network while considering different constraints. The work in [30–35] introduces mixed-integer programming formulations for the optimal placement of multi-controller switches in virtualized Open Flow-enabled SDN networks. The authors in [36–39] address the deployment of multiple controllers that work cooperatively to control a network. The authors proposed a Dynamic Controller Provisioning Problem (DCPP). The DCPP dynamically adapts the number of controllers and their locations with changing network conditions in order to minimize flow setup time and communication overhead. They formulated this problem by using integer linear programming (ILP). So far, the research studies on SDN have been focusing on the controller placement problem, scalability issues, number of controllers and reliability metrics, etc.

The authors of [40–44] address how the centralized paradigm of SDN is a potential vulnerability to the system assuming attackers may launch DDoS attacks against the switches and controllers. The authors further reported that an attacker may create a large number of new flows within a short period of time [21], intending to overwhelm the controller and cause network failure for legitimate users. The authors in [45–47] discuss how DDoS attack

vulnerabilities in Open Flow SDN networks involve overpowering computing or networking resources such that a switch is unable to forward packets as expected. The authors in [48–51] illustrate controller vulnerability to flooding attacks by injecting spoofed request packets continuously; attackers deliberately generate heavy traffic to the controller, causing huge bandwidth occupation in the controller–switch channel, subsequently overloading the flow table in switches. The final goal of attackers is to downgrade or even shutdown the stability and quality of service of the network. Furthermore, the authors introduce a feasible method to protect the network against DDoS attacks more effectively. The authors in [52–56] investigate how an SDN can be utilized to overcome difficulties and effectively block legitimate-looking DDoS attacks mounted by a larger number of bots. Specifically, they discuss a DDoS-blocking application that runs over the SDN controller while using the standard Open Flow interface.

In the work in [50,57–60], some of the authors proposed a novel clustered distributed controller architecture in a real setting of SDN. The distributed cluster implementation comprises multiple popular SDN controllers. The proposed mechanism is evaluated using a real-world network topology running on top of an emulated SDN environment. Their proposed architecture [61–63] is based on distributed controller clustering in SDN that consists of two different types of controllers: an open-source and a commercial-based controllers. Both types of controllers manage different SDN networks. Each controller is set up within a cluster of three nodes; the controllers in each cluster are configured in active mode with one of the controllers acting as the primary controller. The authors of [64–67] address cluster hierarchy and highlight that the implementation of a controller cluster is outside the scope of Open Flow specifications. Their research primarily focused on providing the distributed controllers in SDN and proposed the concept of hierarchy of controllers. However, the hierarchy of controllers and attack-aware logical link assignment needs to be examined to address the single point of failure in SDN networks. In this article, the AALLA model will provide the logical link assignment from networking devices to the cluster (backup) controllers. This mechanism of controller clustering will address the single point of failure under DDoS attacks.

A trust mechanism designed to enhance security protection in SDN-based IoT networks is proposed in [12]. The authors proposed a method to evaluate the trust level of IoT devices based on their operational behaviours and characteristics, allowing the SDN controller to actively monitor and block abnormal devices. The study in [13] presents the DARFESS (Attack-Resilient Framework for Energy Security System), which uses software-defined networking to monitor and control the cyber infrastructure of power systems. Authors provide insights into the security landscape of SDN in [14], aiming to enhance the security posture of SDN deployments and address the evolving security challenges posed by the programmability and centralization of network control. Another study in [15] presents an optimized AI model that effectively detects and mitigates DDoS attacks in SDN environments, showcasing its superiority over traditional machine learning models. The work in [16] emphasizes the significance of a dynamic controller configuration in enhancing the security and resilience of SDN networks, particularly in SCADA systems, and provides insights into the implementation and effectiveness of this approach. The focus of [17] is on the security aspects of distributed SDN controllers in an enterprise SD-WLAN. The authors highlight the security, scalability, reliability, and consistency issues associated with this design.

Currently, research in SDN has been focusing on controller placement strategies, determining the number of controllers, the location for controller placement, and scaling SDN networks. However, failover techniques using attack-aware link assignment methods aiming to mitigate DDoS-based large volumetric attacks have not been investigated thoroughly in SDN, e.g., in [8–19]. This work focuses on tackling a single point of failure and studying the hierarchy of controllers in SDN and presents a novel mathematical model for logical link assignment from switches to the cluster (backup) controllers under DDoS attacks.

So far, none of the attack-aware logical link assignment solutions proposed in the literature have taken into consideration AALLA model formulation. There are also a lot

of works that study controller placement and the division of SDN networks into small domains, but not the formulation scheme that provides a solution for an attack-aware link assignment system in SDN networks under DDoS attacks. The literature review shows that the optimal controller placements only involve one or two input parameters. In the case of placing controllers using clustering, only latency is used to determine controller placement locations. Furthermore, a greedy approach that improves reliability minimizes the failure probability while keeping the shortest distance between the installed controller and switches. A framework that automatically assigns links to switches from the controllers assumes that the controllers are already placed in an SDN network.

3. Attack-Aware Logical Link Assignment (AALLA) Model

This section provides detailed information on the attack-aware link assignment (AALLA) model in integer linear programming (ILP) and its parameters. For the formulation, we assume that the following information is given:

1. The number of switches available in the network and the data packets (traffic) that must be sent to the controller from each switch;
2. The length and the bandwidth available for each link type to be connected between switches and controllers;
3. The characteristics of the different types of controllers. Each type of controller has a cost in USD (\$), number of ports available, maximum number of requests it can handle per second, and the number of available controllers of each type;
4. The maximum link setup latency allowed for switch-to-controller communications. Based on this information, we define the following notation.

To present the notations, we define the set of switches in the network as S . These switches could be of different numbers depending on the size of the network. So, the number of switches is presented as $S = \{s_1, s_2, s_3, \dots, s_n\}$. Each switch can contain a number of packets in it, which are represented by σ^s . Each switch has a cost in USD \$ and this is represented as K^s . We also define the set of controllers in the network and this is represented by C , while the number of the controllers in the set are available as $C = \{c_1, c_2, c_3, \dots, c_n\}$. The cost of these controllers is defined as K^C and is represented in $(c\epsilon C)$ in USD \$. Controllers have different numbers of ports available, represented as α^c , and these are used to connect switches and other controllers. Another characteristic of the controller is that it has processing power μ^c for each controller in the network. There are different types of controllers, which are defined as ∂^C ; some examples are Open Daylight, Floodlight, POX, etc. There are a set of links defined as $L = \{l_1, l_2, l_3, \dots, l_n\}$ and these links have some characteristics, for example ω^l shows the bandwidth of the link of type $(l\epsilon L)$ and ϕ^l shows the price of the link of type $(l\epsilon L)$ in USD \$.

In another scenario, we define the notations for the backup system, such as the set of backup controllers in the network as $BC = \{bc_1, bc_2, bc_3, \dots, bc_n\}$. The cost of the backup controller of type $(bc\epsilon BC)$ in USD \$ is defined by K^{bc} and the number of ports available in the backup controller $(bc\epsilon BC)$ are shown by ∂^{bc} . Similarly the processing power of the backup controller $(bc\epsilon BC)$ is shown as μ^{bc} and the number of backup (cluster) controllers of type $(bc\epsilon BC)$ are defined as ∂^{bc} . Finally, the DDoS attacks are defined as $DDoS^s$, which shows if the resultant number is 1 that an attack happened on a switch and if its 0 no attacks happened at all.

The proposed AALLA model has some static notations as well, which are set in the model file of the AMPL IDE setup. The maximum delay allowed in the network for flow-setup latencies is denoted by λ and the data packet size for each packet in bytes is denoted by β . " t " is the speed of light and/or communication channel. It can vary as per the medium of communication, i.e., wired or wireless. There is a function that converts the bandwidth of the link into byte/s, which is denoted by B^{byte} .

The proposed AALLA mathematical model includes the following decision variables:

$$V^{1sc} = \begin{cases} 1, & \text{if link of type (leL) is installed} \\ & \text{between switch (seS)} \\ & \text{and controller (ceC);} \\ 0, & \text{otherwise.} \end{cases}$$

$$W^{1cj} = \begin{cases} 1, & \text{if link of type (leL) is installed} \\ & \text{between controller (ceC)} \\ & \text{and controller (jeC);} \\ 0, & \text{otherwise.} \end{cases}$$

$$X^{1cbc} = \begin{cases} 1, & \text{if link of type (leL) is} \\ & \text{installed between} \\ & \text{controller (ceC) and} \\ & \text{backup controller (bceBC);} \\ 0, & \text{otherwise.} \end{cases}$$

$$Y^{llsbc} = \begin{cases} 1, & \text{if logical link of type (lleLL)} \\ & \text{is installed between} \\ & \text{switch (seS) and} \\ & \text{backup controller (bceBC);} \\ 0, & \text{otherwise.} \end{cases}$$

$$Z^{1sv} = \begin{cases} 1, & \text{if link of type (leL) is installed} \\ & \text{between switch (seS)} \\ & \text{and switch (veS);} \\ 0, & \text{otherwise.} \end{cases}$$

3.1. Cost Function for AALLA Mathematical Model

The objective of the AALLA model is to optimize the cost of deploying a secure SDN network at the planning stage. This model's development factored in the cost of controllers, switches, and links. The function space (a, b) in the following equations will calculate the distance between two points as point a and point b, where a and b refer to switches.

Equations (1) and (2) are the cost of controllers, $Cost^c(k)$, and cluster (backup) controllers, $Cost^{bc}(k)$, for SDN network deployment.

$$Cost^c(k) = \sum_{ceC} K^C \quad (1)$$

$$Cost^{bc}(k) = \sum_{bceBC} K^{bc} \quad (2)$$

Equation (3) is the cost of the switches, $Cost^s(v)$, connecting to the controllers.

$$Cost^s(v) = \sum_{seS} K^S \sum_{leL} \sum_{ceC} V^{1sc} \quad (3)$$

Equations (4)–(6) calculate the cost of links for connecting switches to the controller, $Cost^l(v)$; for connecting a controller to a controller, $Cost^l(w)$; and for connecting a controller to cluster (backup) controllers, $Cost^l(x)$.

$$Cost^l(v) = \sum_{leL} \phi^1 \sum_{seS} \sum_{ceC} \text{space}(s, c) V^{1sc} \quad (4)$$

$$Cost^l(w) = \sum_{leL} \phi^1 \sum_{ceC} \sum_{\substack{jeC \\ c < j}} \text{space}(c, j) W^{1cj} \quad (5)$$

$$\text{Cost}^l(x) = \sum_{l \in L} \phi^l \sum_{c \in C} \sum_{bc \in BC} \text{space}(c, bc) X^{lcbc} \quad (6)$$

Equation (7) is the cost of links, $\text{Cost}^l(z)$, connecting switches together.

$$\text{Cost}^l(z) = \sum_{l \in L} \phi^l \sum_{\substack{se \in S \\ ve \in S \\ s < v}} \text{space}(s, v) Z^{lsv} \quad (7)$$

The following equation is the cost of logical links, $\text{Cost}^{ll}(y)$, connecting switches to the cluster (backup) controller under a DDoS attack with higher processing power of cluster (backup) controllers. This is the operating cost of the SDN network at the planning stage.

$$\text{Cost}^{ll}(y) = \sum_{ll \in LL} \sum_{se \in S} \sum_{bc \in BC} \left(\frac{1}{\mu^{bc}} \right) Y^{llsbc} \quad (8)$$

3.2. The AALLA Model (ILP) Formulation

Our formulation for the AALLA problem can be derived as described in this subsection. Firstly, to minimize the cost of controllers and backup controllers, the cost of links connecting switches together, the cost of switches, and the cost of linking controllers together, we devise the following Equation (9).

$$(\text{Cost}^c(k) + \text{Cost}^{bc}(k) + \text{Cost}^s(v) + \text{Cost}^l(v) + \text{Cost}^l(w) + \text{Cost}^l(x) + \text{Cost}^{ll}(y) + \text{Cost}^l(z)) \quad (9)$$

This section also provides the equations that are the core part of the model to minimize the cost of SDN network deployment. Namely, the following constraint ensures that the controller has a sufficient number of ports to connect switches and other controllers.

$$\sum_{c \in C} \sum_{l \in L} W^{lj} + \sum_{se \in S} \sum_{l \in L} V^{lsc} \leq \alpha^c \quad (j \in C) \quad (10)$$

Moreover, the next constraint ensures that the link chosen between the controller and the switch can handle the bandwidth needed by the switch.

$$(\sigma^s, \beta) \sum_{l \in L} V^{lsc} \leq \sum_{l \in L} f(\omega^l) V^{lsc} \quad (se \in S, ce \in C) \quad (11)$$

Constraint (12) is obtained for the case where the round-trip flow-setup latencies for unmatched flows in each of the switches is set below or equal to λ .

$$(\beta / \omega^l) V^{lsc} \leq \lambda \quad (ce \in C, le \in L, se \in S) \quad (12)$$

Constraints (13) and (14) ensure all switches are connected to controllers. Constraint number (15) makes sure the switches are interconnected with their respective controller only.

$$\sum_{l \in L} \sum_{ce \in C} V^{lsc} \geq 1 \quad (se \in S) \quad (13)$$

$$\sum_{l \in L} Z^{lsv} \leq 1 \quad (se \in S, ve \in S, s < v) \quad (14)$$

$$\sum_{l \in L} V^{lsc} + \sum_{l \in L} V^{lvc} \leq \sum_{l \in L} Z^{lsv} + 1 \quad (j \in C, ve \in S, se \in S, s < v) \quad (15)$$

The constraints in Equations (16)–(18) ensure all the controllers are interconnected together, and each controller is connected to all cluster (backup) controllers using full-mesh

topology (ensuring that the links from the controller to the cluster are being assigned in one direction only).

$$\sum_{l \in L} W^{lcj} \geq 1 \quad (c \in C, j \in C, c < j) \quad (16)$$

$$\sum_{l \in L} \sum_{c \in C} X^{lcbc} \geq \sum_{l \in L} \sum_{c \in C} \sum_{\substack{j \in C \\ c < j}} W^{lcj} / 2 \quad (k \in BC) \quad (17)$$

$$\sum_{l \in L} X^{lcbc} \geq 1 \quad (j \in C, k \in BC) \quad (18)$$

The constraint number in Equation (19) ensures that the number of data packets that each switch sends can be processed by the controller.

$$\sum_{l \in L} \sum_{s \in S} \sigma^s V^{lsc} \leq \mu^c \quad (c \in C) \quad (19)$$

The constraints in (20), (21), and (22) are for the scenario where the affected switches are connected to only one cluster (backup) controller (bceBC) under a DDoS attack using the logical links, and the processing power of the cluster (backup) controller is equal or at least higher than the original controller.

$$\sum_{l \in LL} \sum_{j \in BC} Y^{llsbc} \leq 1 \quad (s \in S) \quad (20)$$

$$\sum_{l \in LL} \sum_{j \in BC} Y^{llsbc} \geq DDoS^s \quad (s \in S) \quad (21)$$

$$\sum_{l \in L} \sum_{l \in LL} \sum_{c \in C} \sum_{s \in S} V^{lsc} * Y^{llsbc} * \mu^c \leq \mu^{bc} \quad (bceBC) \quad (22)$$

4. Experimental Results and Discussion

In this section, we discuss the experimental results and simulation platform tools used for AALLA mathematical model formulation in detail.

A mathematical programming language (AMPL) was used to formulate the AALLA model along with the IBM ILOG CPLEX 12.7.0.0: optimal integer solution; this is a powerful solver for AMPL code execution. In our experiments, we used an Acer Aspire XC-780 workstation, Intel® Core™ i7-6700 x64-based 6th-generation CPU @ 3.40GHz, with a memory of 8 GB RAM and virtual memory of 128 GB on Windows 10 x64.

Table 1a,b present the two scenarios, known as problem (A) and problem (B), used for the AALLA model simulation. They consist of the following elements as depicted in Figures 1 and 2.

Table 1. The input dataset used for AALLA model.

Controllers for AALLA Model—Problem (a)				
Controllers	Alpha_c (α^c)	Mu_c (μ^c)	Kappa_c (K^C)	Phi_c (∂^C)
C1	8	8000	7000 USD	6
C2	16	9000	8000 USD	7
C3	32	10,000	9000 USD	4
Cluster (backup) Controllers for AALLA Model				
Controllers	Alpha_bc (α^{bc})	Mu_bc (μ^{bc})	Kappa_bc (K^{bc})	Phi_bc (∂^{bc})
BC1	16	9900	10,000 USD	6
BC2	32	11,000	10,500 USD	7

Table 1. Cont.

Controllers for AALLA Model—Problem (a)			
Links for AALLA Model			
Links	Omega_l (ω^l)	Phi_l (ϕ^l)	
L1	10,000,000	0.28 USD	
L2	200,000,000	0.34 USD	
L3	3,000,000,000	6 USD	
Switches for AALLA Model			
Switches	Sigma_s (σ^s)	Kappa_s (K^s)	
S1	100	0.10 USD	
S2	200	0.15 USD	
S3	300	0.20 USD	
S4	400	0.30 USD	
S5	500	0.40 USD	
S6	600	0.50 USD	
Other inputs to the AALLA Model			
Input type	Symbol	Data/Units	
Data packet size	β	1700 bytes	
Function for bandwidth conversion (per second)	B^{byte}	1/8 per second	
Distance between two points	space	200 m	
Maximum delay	λ	349 ms	
Average time	δ	0.001 ms	
Speed of communication channel (wired or wireless)	t	0.59 per second	
Traffic intensity	p	2100 total # packets	
Controllers for AALLA Model—Problem (b)			
Controllers	Alpha_c (α^c)	Mu_c (μ^c)	Kappa_c (K^c)
C1	8	8000	7000 USD
C2	16	9000	8000 USD
C3	32	10,000	9000 USD
Cluster (backup) Controllers for AALLA Model			
Controllers	Alpha_bc (α^{bc})	Mu_bc (μ^{bc})	Kappa_bc (K^{bc})
BC1	16	9900	10,000 USD
BC2	32	11,000	10,500 USD
Links for AALLA Model			
Links	Omega_l (ω^l)	Phi_l (ϕ^l)	
L1	10,000,000	0.28 USD	
L2	200,000,000	0.34 USD	
L3	3,000,000,000	6 USD	
Switches for AALLA Model			
Switches	Sigma_s (σ^s)	Kappa_s (K^s)	
S1	100	0.10 USD	
S2	200	0.15 USD	
S3	300	0.20 USD	
S4	400	0.30 USD	
S5	500	0.40 USD	
S6	600	0.50 USD	

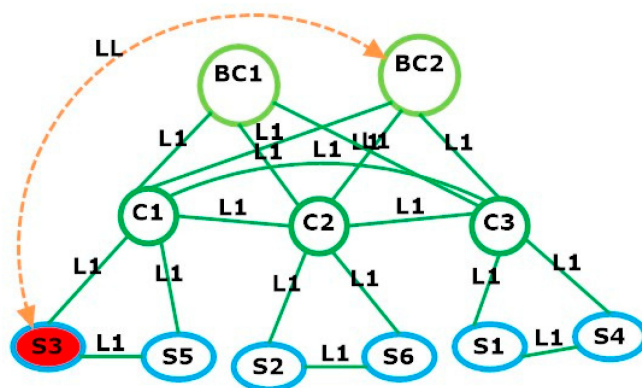


Figure 2. Logical link assignment between switch S3 and cluster controller BC2 under DDoS attack. Red nodes are used for nodes under attack and orange links are used to denote that a compromised switch has been restored using the next available logical link.

Three controllers for problems (A) and (B) are given as C1, C2, and C3 with different specifications and a cost in USD.

Two cluster (backup) controllers BC1 and BC2 for problem (A) and problem (B) are used and they will be activated upon DDoS attack occurred on the switches connected with C1, C2, and C3.

Three type of links, L1, L2, L3, are used with different prices in USD and bandwidth in bytes, respectively. Six input switches, as S1, S2, S3, S4, S5, S6, for problem (A) and the same for problem (B) are used with a price in USD.

Some of the other constants used in this experimental setup are as follows: Beta is a constant in the model file, which is set for the size of data packets in bytes. A function of B^{byte} is used as a source of converting the GBs/MBs into Bytes per second. Space/range is a function that is used to calculate the distance between two points such as point A and point B. The maximum delay is set in the model using λ , which is allowed for the flow-setup latency in the network. Delta “ δ ” is used for the average time in milliseconds for processing a packet in switches and “ t ” is used for the speed of the medium of communication, such as wired or wireless network.

The simulation results are described and presented in Table 2. Here, we can observe that the total data packets are 2100 p, processed for both problems (a) and (b) with 1398 and 2986 CPLEX iterations, respectively. The DDoS attack happened on switch S3 in problem (a) in Figure 2 and switches S3 and S6 in problem (b), as illustrated in Figure 3, while a minimized cost of 45,509 USD is incurred for SDN planning.

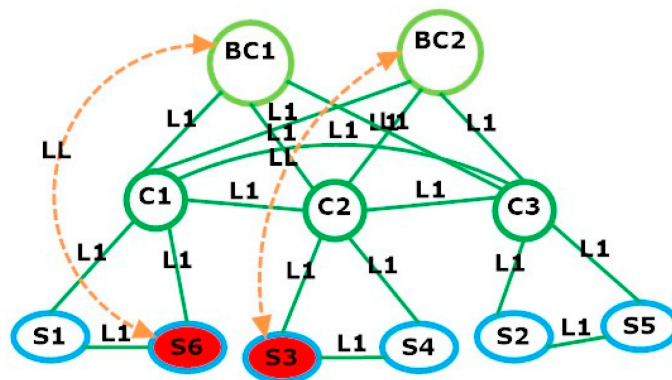


Figure 3. Logical link assignment from switch S3 to BC2 and from switch S6 to BC1 cluster (backup) controller under DDoS attack. Red nodes are used for nodes under attack and orange links are used to denote that a compromised switch has been restored using the next available logical link.

Table 2. AMPL and CPLEX solutions for AALLA model in SDN.

(a) Input Dataset # 1 Results Using AMPL and CPLEX Solver								
Switch	Link	Controller	Cluster (Backup) Controller	Switch Assigned to Cluster Controller	Data Size (p)	Cost (USD)	CPLEX Iterations	DDoS Attack Location
S1	L1	C1	BC1	S3	2100 p	45,509	1398	no attack
S2	L2	C2	BC2					S3
S3	L3	C3						no attack
S4								no attack
S5								no attack
S6								no attack
(b) Input Dataset # 2 Results Using AMPL & CPLEX Solver								
Switch	Link	Controller	Cluster (Backup) Controller	Switch Assigned to Cluster Controller	Data size (p)	Cost (USD)	CPLEX Iterations	DDoS Attack Location
S1	L1	C1	BC1	S3	2100 p	45,509	2986	no attack
S2	L2	C2	BC2	S6				S3
S3	L3	C3						no attack
S4								no attack
S5								no attack
S6								S6

As per the results obtained from AMPL simulation, we observed that multiple DDoS attacks in SDN networks may incur more cost in terms of restoring the networking devices to the cluster controller. The reasons are that the logical links will be chosen upon the basis of processing power; therefore, if an attack happens on a switch, then the model will choose the higher-processing-power cluster (backup) controller in order to restore the network services [68–70]. The results indicate that the model can be used to plan small- and medium-scale enterprises (SMEs) in SDN networks to reduce the impact of DDoS attacks and to failover a single point of failure in the SDN with optimal cost for deployment.

The experimental results provide insights into the dynamics of SDN environments, particularly in the context of mitigating DDoS attacks. One of the key findings of our study is that the occurrence of DDoS attacks within SDN networks can significantly escalate the cost associated with restoring networking devices to their respective cluster controllers. This observation highlights the critical need for proactive measures to defend against and recover from such attacks. The rationale behind the increased cost is rooted in the model's logic, which prioritizes processing power when selecting logical links for network restoration. In the case of a DDoS attack targeting a switch, the model chooses the cluster controller with higher processing power to ensure the efficient restoration of network services. While this approach is indeed effective in terms of ensuring network resilience, it comes at an increased financial cost. This insight highlights the trade-off between network robustness and cost, a critical consideration for network administrators and decision-makers.

The findings of this research align with previous studies (e.g., [68–70]). The ability to model and simulate such scenarios using mathematical optimization techniques, as demonstrated in this study, provides a powerful tool for network design and operation. By using the AALLA model, small- and medium-sized enterprises (SMEs) can strategically plan their SDN networks to reduce the impact of DDoS attacks and implement cost-effective failover mechanisms.

In conclusion, our experimental results shed light on the complex interplay between DDoS attacks, network resilience, and cost considerations in SDN environments. Our model provides a tool that can be used at the planning stage of an SDN network to provide proactive defence strategies to mitigate the financial and operational consequences of

DDoS attacks. The AALLA model offers a promising avenue for optimizing SDN network deployment and managing the risks associated with DDoS attacks, ultimately enhancing the overall reliability and security of modern network infrastructures.

5. Conclusions

In this paper, we have proposed a novel AALLA mathematical model for the attack-aware link assignment problem between the switches and cluster (backup) controllers in SDN networks. Given the set of switches in the SDN network that must be managed by the controller(s), the proposed model simultaneously determined the optimal bandwidth for the links, the assignment of the logical links to the cluster (backup) controllers under DDoS attack, as well as the interconnections between all the network elements to minimize the SDN deployment cost at the planning stage. Our simulation results have shown that this linear model performed well for the SDN network under DDoS attacks to avoid a single point of failure. We tested two input datasets with multiple attacks to analyse the results. The outcome of two problem sizes have shown that the DDoS-affected switch in scenario (a) is switch S3 and in scenario (b) the switches are S3 and S6, which were assigned to the cluster (backup) controller using logical links. This method provides the SDN network with high availability, reliability, and uninterrupted services to fulfil internet service providers (ISPs) and end-user requirements. Our plans for future work include the validation of the proposed AALLA model in real-world SDN environments. This could involve collaborating with industry partners or deploying the solutions in testbeds to assess their practicality, scalability, and performance, improving the detection and mitigation techniques for DDoS attacks in SDN networks. Also, we plan to investigate and develop more advanced controller clustering methods to enhance the resilience of SDN networks against DDoS attacks. This may include extending the current model and exploring optimal strategies for load balancing, fault tolerance, and scalability in controller clusters.

Author Contributions: Conceptualization, S.A.; Validation, M.R.H.; Formal analysis, M.R.H.; Investigation, S.A.; Data curation, M.R.H.; Writing – original draft, S.A. and M.R.H.; Writing – review & editing, S.A., A.M. and N.P.; Supervision, S.C.T., C.K.L. and Z.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This research work was fully supported by the research grant of TM R&D and Multimedia University (MMU), Cyberjaya, Malaysia. We are very thankful to the team at TM R&D and Multimedia University (MMU) for providing generous support to our research studies.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rawat, D.B.; Reddy, S.R. Software defined networking architecture, security and energy efficiency: A survey. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 325–346. [[CrossRef](#)]
2. Shin, S.; Gu, G. Attacking software-defined networks: A first feasibility study. In Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, Hong Kong, China, 16 August 2013; pp. 165–166.
3. Cox, J.H.; Chung, J.; Donovan, S.; Ivey, J.; Clark, R.J.; Riley, G.; Owen, H.L. Advancing Software-Defined Networks: A Survey. *IEEE Access* **2017**, *5*, 25487–25526. [[CrossRef](#)]
4. Lange, S.; Gebert, S.; Zinner, T.; Tran-Gia, P.; Hock, D.; Jarschel, M.; Hoffmann, M. Heuristic approaches to the controller placement problem in large scale SDN networks. *IEEE Trans. Netw. Serv. Manag.* **2015**, *12*, 4–17. [[CrossRef](#)]
5. Yeganeh, S.; Ganjali, Y. Kandoo: A framework for efficient and scalable offloading of control applications. In Proceedings of the ACM SIGCOMM Hot Topics in Software Defined Networking (HotSDN), Helsinki, Finland, 13 August 2012.
6. Casado, M. Scalability and reliability of logically centralized controller. In Proceedings of the Stanford CIO Summit, Stanford, CA, USA, 15 June 2010.
7. Shu, Z.; Wan, J.; Li, D.; Lin, J.; Vasilakos, A.V.; Imran, M. Security in software-defined networking: Threats and countermeasures. *Mob. Netw. Appl.* **2016**, *21*, 764–776. [[CrossRef](#)]
8. Shohani, R.B.; Mostafavi, S.A. Introducing a new linear regression based method for early DDoS attack detection in SDN. In Proceedings of the 2020 6th International Conference on Web Research (ICWR), Tehran, Iran, 22–23 April 2020; pp. 126–132.

9. Sufiev, H.; Haddad, Y. DCF: Dynamic cluster flow architecture for SDN control plane. In Proceedings of the 2017 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 8–10 January 2017; pp. 172–173.
10. Bouzidi, E.H.; Outtagarts, A.; Langar, R.; Boutaba, R. Dynamic clustering of software defined network switches and controller placement using deep reinforcement learning. *Comput. Netw.* **2022**, *207*, 108852. [[CrossRef](#)]
11. Macedo, R.; de Castro, R.; Santos, A.; Ghamri-Doudane, Y.; Nogueira, M. Self-Organized SDN Controller Cluster Conformations Against DDoS Attacks Effects. In Proceedings of the Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–6.
12. Tsai, P.W.; Lee, C.W.; Wang, T.W. Design and Development of a Trust Mechanism to Enhance Security Protection on SDN-based IoT Network. In Proceedings of the 2023 24th Asia-Pacific Network Operations and Management Symposium (APNOMS), Detroit, MI, USA, 21–23 September 2023; pp. 125–130.
13. Jin, D.; Qu, Y.; Liu, X.; Hannon, C.; Yan, J.; Aved, A.J.; Morrone, P. Dynamic Data-Driven Approach for Cyber-Resilient and Secure Critical Energy Systems. In *Handbook of Dynamic Data Driven Applications Systems*; Springer International Publishing: Cham, Switzerland, 2023; Volume 2, pp. 807–831.
14. Bhuiyan, Z.A.; Islam, S.; Islam, M.M.; Ullah, A.A.; Naz, F.; Rahman, M.S. On the (in) Security of the Control Plane of SDN Architecture: A Survey. *IEEE Access* **2023**, *11*, 91550–91582. [[CrossRef](#)]
15. Al-Dunainawi, Y.; Al-Kaseem, B.R.; Al-Raweshidy, H.S. Optimized Artificial Intelligence Model for DDoS Detection in SDN Environment. *IEEE Access* **2023**, *11*, 106733–106748. [[CrossRef](#)]
16. DeLany, R.; Smith, A.; Li, Y.; Du, L. SDN Dynamic Controller Configuration to Mitigate Compromised Controllers. In Proceedings of the 2023 IEEE Transportation Electrification Conference & Expo (ITEC), Detroit, MI, USA, 21–23 June 2023; pp. 1–5.
17. Shaji, N.S.; Muthalagu, R. Survey on security aspects of distributed software-defined networking controllers in an enterprise SD-WLAN. *Digit. Commun. Netw.* **2023**. [[CrossRef](#)]
18. Lemeshko, O.; Yeremenko, O.; Mersni, A.; Gazda, J. Improvement of Confidential Messages Secure Routing over Paths with Intersection in Cyber Resilient Networks. In Proceedings of the 2022 XXVIII International Conference on Information, Communication and Automation Technologies (ICAT), Sarajevo, Bosnia and Herzegovina, 16–18 June 2022; pp. 1–6.
19. Lemeshko, O.; Yeremenko, O.; Yevdokymenko, M.; Shapovalova, A.; Baranovskyi, O. Complex investigation of the compromise probability behavior in traffic engineering oriented secure routing model in software-defined networks. In *Future Intent-Based Networking: On the QoS Robust and Energy Efficient Heterogeneous Software Defined Networks*; Springer International Publishing: Cham, Switzerland, 2021; pp. 145–160.
20. Yan, Q.; Yu, F.R.; Gong, Q.; Li, J. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 602–622. [[CrossRef](#)]
21. Mallikarjunan, K.N.; Muthupriya, K.; Shalinie, S.M. A survey of distributed denial of service attack. In Proceedings of the 2016 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 7–8 January 2016; pp. 1–6.
22. Xu, Y.; Liu, Y. DDoS attack detection under SDN context. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
23. Wang, T.; Liu, F.; Guo, J.; Xu, H. Dynamic sdn controller assignment in data center networks: Stable matching with transfers. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
24. Dvir, A.; Haddad, Y.; Zilberman, A. The controller placement problem for wireless SDN. *Wirel. Netw.* **2019**, *25*, 4963–4978. [[CrossRef](#)]
25. Müller, L.F.; Oliveira, R.R.; Luizelli, M.C.; Gaspary, L.P.; Barcellos, M.P. Survivor: An enhanced controller placement strategy for improving SDN survivability. In Proceedings of the Global Communications Conference (GLOBECOM), Austin, TX, USA, 8–12 December 2014; pp. 1909–1915.
26. Muqaddas, A.S.; Bianco, A.; Giaccone, P.; Maier, G. Inter-controller traffic in ONOS clusters for SDN networks. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6.
27. Luo, M.; Li, Q.; Bo, M.; Lin, K.; Wu, X.; Li, C.; Lu, S.; Chou, W. Design and implementation of a scalable sdn-of controller cluster. In Proceedings of the INFOCOMP 2015, Brussels, Belgium, 21–26 June 2015; p. 55.
28. Sallahi, A.; St-Hilaire, M. Optimal model for the controller placement problem in software defined networks. *IEEE Commun. Lett.* **2015**, *19*, 30–33. [[CrossRef](#)]
29. Zilberman, A.; Haddad, Y.; Erlich, S.; Peretz, Y.; Dvir, A. SDN Wireless Controller Placement Problem—The 4G LTE-U Case. *IEEE Access* **2021**, *9*, 16225–16238. [[CrossRef](#)]
30. Blenk, A.; Basta, A.; Zerwas, J.; Reisslein, M.; Kellerer, W. Control plane latency with sdn network hypervisors: The cost of virtualization. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 366–380. [[CrossRef](#)]
31. Karakus, M.; Duresi, A. A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN). *Comput. Netw.* **2017**, *112*, 279–293. [[CrossRef](#)]
32. Wang, G.; Zhao, Y.; Huang, J.; Wang, W. The controller placement problem in software defined networking: A survey. *IEEE Netw.* **2017**, *31*, 21–27. [[CrossRef](#)]
33. Samir, M.; Azab, M.; Samir, E. SD-CPC: SDN Controller Placement Camouflage based on Stochastic Game for Moving-target Defense. *Comput. Commun.* **2021**, *168*, 75–92. [[CrossRef](#)]

34. Hu, Y.; Wendong, W.; Gong, X.; Que, X.; Shiduan, C. Reliability-aware controller placement for software-defined networks. In Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), Ghent, Belgium, 27–31 May 2013; pp. 672–675.
35. Li, X.; Tang, F.; Fu, L.; Yu, J.; Chen, L.; Liu, J.; Zhu, Y.; Yang, L.T. Optimized controller provisioning in software-defined LEO satellite networks. *IEEE Trans. Mob. Comput.* **2022**, *22*, 4850–4864. [[CrossRef](#)]
36. Bari, M.F.; Roy, A.R.; Chowdhury, S.R.; Zhang, Q.; Zhani, M.F.; Ahmed, R.; Boutaba, R. Dynamic controller provisioning in software defined networks. In Proceedings of the 2013 9th International Conference on Network and Service Management (CNSM), Zurich, Switzerland, 14–18 October 2013; pp. 18–25.
37. Han, Z.; Xu, C.; Xiong, Z.; Zhao, G.; Yu, S. On-Demand Dynamic Controller Placement in Software Defined Satellite-Terrestrial Networking. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 2915–2928. [[CrossRef](#)]
38. Das, T.; Gurusamy, M. Controller placement for resilient network state synchronization in multi-controller sdn. *IEEE Commun. Lett.* **2020**, *24*, 1299–1303. [[CrossRef](#)]
39. Heller, B.; Sherwood, R.; McKeown, N. The controller placement problem. In Proceedings of the First Workshop on Hot Topics in Software Defined Networks, Helsinki, Finland, 13 August 2012; pp. 7–12.
40. Wei, L.; Fung, C. FlowRanger: A request prioritizing algorithm for controller DoS attacks in software defined networks. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 5254–5259.
41. Balarezo, J.F.; Wang, S.; Chavez, K.G.; Al-Hourani, A.; Kandeepan, S. A survey on DoS/DDoS attacks mathematical modelling for traditional, SDN and virtual networks. *Eng. Sci. Technol. Int. J.* **2022**, *31*, 101065. [[CrossRef](#)]
42. Scaranti, G.F.; Carvalho, L.F.; Barbon, S.; Proença, M.L. Artificial Immune Systems and Fuzzy Logic to Detect Flooding Attacks in Software-Defined Networks. *IEEE Access* **2020**, *8*, 100172–100184. [[CrossRef](#)]
43. Ali, T.E.; Chong, Y.W.; Manickam, S. Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review. *Appl. Sci.* **2023**, *13*, 3183. [[CrossRef](#)]
44. Aladaileh, M.A.; Anbar, M.; Hasbullah, I.H.; Chong, Y.W.; Sanjalawe, Y.K. Detection Techniques of Distributed Denial of Service Attacks on Software-Defined Networking Controller—A Review. *IEEE Access* **2020**, *8*, 143985–143995. [[CrossRef](#)]
45. Kandoi, R.; Antikainen, M. Denial-of-service attacks in OpenFlow SDN networks. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 1322–1326.
46. Yan, Q.; Yu, F.R. Distributed denial of service attacks in software-defined networking with cloud computing. *IEEE Commun. Mag.* **2015**, *53*, 52–59. [[CrossRef](#)]
47. Yonghong, F.; Jun, B.; Jianping, W.; Ze, C.; Ke, W.; Min, L. A dormant multi-controller model for software defined networking. *China Commun.* **2014**, *11*, 45–55. [[CrossRef](#)]
48. Dao, N.N.; Park, J.; Park, M.; Cho, S. A feasible method to combat against DDoS attack in SDN network. In Proceedings of the 2015 International Conference on Information Networking (ICOIN), Siem Reap, Cambodia, 12–14 January 2015; pp. 309–311.
49. Saxena, U.; Sodhi, J.S.; Singh, Y. An Analysis of DDoS Attacks in a Smart Home Networks. In Proceedings of the 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 29–31 January 2020; pp. 272–276.
50. Erhan, D.; Anarim, E.; Kurt, G.K. DDoS attack detection using matching pursuit algorithm. In Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU), Zonguldak, Turkey, 16–19 May 2016; pp. 1081–1084.
51. Huang, K.; Yang, L.X.; Yang, X.; Xiang, Y.; Tang, Y.Y. A low-cost distributed denial-of-service attack architecture. *IEEE Access* **2020**, *8*, 42111–42119. [[CrossRef](#)]
52. Lim, S.; Ha, J.; Kim, H.; Kim, Y.; Yang, S. A SDN-oriented DDoS blocking scheme for botnet-based attacks. In Proceedings of the 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN), Shanghai, China, 8–11 July 2014; pp. 63–68.
53. Wang, B.; Zheng, Y.; Lou, W.; Hou, Y.T. DDoS attack protection in the era of cloud computing and software-defined networking. *Comput. Netw.* **2015**, *81*, 308–319. [[CrossRef](#)]
54. Wang, H.; Xu, L.; Gu, G. Floodguard: A dos attack prevention extension in software-defined networks. In Proceedings of the 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Rio de Janeiro, Brazil, 22–25 June 2015; pp. 239–250.
55. Haider, S.; Akhunzada, A.; Mustafa, I.; Patel, T.B.; Fernandez, A.; Choo, K.K.R.; Iqbal, J. A deep cnn ensemble framework for efficient ddos attack detection in software defined networks. *IEEE Access* **2020**, *8*, 53972–53983. [[CrossRef](#)]
56. Pérez-Díaz, J.A.; Valdovinos, I.A.; Choo, K.K.R.; Zhu, D. A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. *IEEE Access* **2020**, *8*, 155859–155872. [[CrossRef](#)]
57. Abdelaziz, A.; Fong, A.T.; Gani, A.; Garba, U.; Khan, S.; Akhunzada, A.; Talebian, H.; Choo, K.K.R. Distributed controller clustering in software defined networks. *PLoS ONE* **2017**, *12*, e0174715. [[CrossRef](#)]
58. Wang, S.; Balarezo, J.F.; Chavez, K.G.; Al-Hourani, A.; Kandeepan, S.; Asghar, M.R.; Russello, G. Detecting flooding DDoS attacks in software defined networks using supervised learning techniques. *Eng. Sci. Technol. Int. J.* **2022**, *35*, 101176. [[CrossRef](#)]
59. Singh, M.P.; Bhandari, A. New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges. *Comput. Commun.* **2020**, *154*, 509–527. [[CrossRef](#)]
60. Li, J.; Tu, T.; Li, Y.; Qin, S.; Shi, Y.; Wen, Q. DoSGuard: Mitigating denial-of-service attacks in software-defined networks. *Sensors* **2022**, *22*, 1061. [[CrossRef](#)] [[PubMed](#)]

61. Gurusamy, U.; Hariharan, K.; Manikandan, M.S.K. Path optimization of box-covering based routing to minimize average packet delay in software defined network. *Peer-to-Peer Netw. Appl.* **2020**, *13*, 932–939. [[CrossRef](#)]
62. Wang, X.; Yang, Y.; Liu, H.; Ren, J.; Xu, S.; Wang, S.; Yu, S. Efficient measurement of round-trip link delays in software-defined networks. *J. Netw. Comput. Appl.* **2020**, *150*, 102468. [[CrossRef](#)]
63. Parashar, M.; Poonia, A.; Satish, K. A Survey of Attacks and their Mitigations in Software Defined Networks. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–8.
64. Goransson, P.; Black, C.; Culver, T. *Software Defined Networks: A Comprehensive Approach*; Morgan Kaufmann: Burlington, MA, USA, 2016.
65. Ros, F.J.; Ruiz, P.M. On reliable controller placements in software-defined networks. *Comput. Commun.* **2016**, *77*, 41–51. [[CrossRef](#)]
66. Rasol, K.A.R.; Domingo-Pascual, J. Joint Latency and Reliability-Aware Controller Placement. In Proceedings of the 2021 International Conference on Information Networking (ICOIN), Jeju Island, Republic of Korea, 13–16 January 2021; pp. 197–202.
67. Hock, D.; Hartmann, M.; Gebert, S.; Jarschel, M.; Zinner, T.; Tran-Gia, P. Pareto-optimal resilient controller placement in SDN-based core networks. In Proceedings of the 2013 25th International Teletraffic Congress (ITC), Shanghai, China, 10–12 September 2013; pp. 1–9.
68. Yazici, V.; Sunay, M.O.; Ercan, A.O. Controlling a software-defined network via distributed controllers. *arXiv* **2014**, arXiv:1401.7651.
69. Jalili, A.; Keshtgari, M.; Akbari, R. A new framework for reliable control placement in software-defined networks based on multi-criteria clustering approach. *Soft Comput.* **2020**, *24*, 2897–2916. [[CrossRef](#)]
70. Latah, M.; Toker, L. Load and stress testing for SDN's northbound API. *SN Appl. Sci.* **2020**, *2*, 122. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.