https://doi.org/10.52326/jes.utm.2021.28(4).08
CZU 004.7:004.4

ASSURING THE SDN SECURITY BY MODELLING AND COMPARING SDN PROPOSED TOPOLOGIES USING PETRI NETS

Ali Ameen, ORCID ID: 0000-0002-5451-8257

Technical University of Moldova, 168 Stefan cel Mare Blvd., MD-2004 Chisinau, Republic of Moldova
Corresponding author: Ali Ameen, *alisalmanhussein@yahoo.com*

Received: 09. 04. 2021
Accepted: 11. 12. 2021

Abstract. The soaring number of applications for autonomous systems in different aspects like air, sea, and space is creating the need for new methodologies and architectures' technologies to consolidate the verification of system-level and system-of-systems level. The implementation of cybersecurity standards and software is critical to supporting infrastructure. This article discusses some security issues regarding autonomous systems' computer networks. It proposes the usage of Software-Defined Networks (SDN) technologies as a solution, after providing better security in SDN environment through the usage of the HYDRA framework and the usage of multiple controllers in specific topologies to ensure the security of SDN in precise and to ensure the security of the autonomous systems' computer networks in general as well. We propose a framework that contains 3 different types of controllers' topologies and each topology can use 4 algorithms, HYDRA, VPN, Double RSA, and least but not last comes blockchain technology which is the core of our security.

Keywords: *autonomous, systems, computer, network, security, SDN, technology, controller, topology, algorithm.*

Rezumat. Numărul tot mai mare de aplicații pentru sisteme autonome în diferite aspecte precum aerul, marea și spațiul creează nevoia de noi metodologii și tehnologii de arhitectură pentru a consolida verificarea la nivel de sistem și la nivel de sistem de sisteme. Implementarea standardelor și a software-ului de securitate cibernetică este esențială pentru sprijinirea infrastructurii. Acest articol discută unele probleme de securitate privind rețelele de calculatoare ale sistemelor autonome. Acesta propune utilizarea tehnologiilor Software-Defined Networks (SDN) ca soluție, după ce a asigurat o mai bună securitate în mediul SDN prin utilizarea cadrului HYDRA și utilizarea mai multor controlere în topologii specifice pentru a asigura securitatea SDN în mod precis și pentru asigurarea securității rețelelor de calculatoare ale sistemelor autonome și în general. Propunem un cadru care conține 3 tipuri diferite de topologii de controlere și fiecare topologie poate folosi 4 algoritmi, HYDRA, VPN, Double RSA, și cel mai puțin, dar nu ultimul, este tehnologia blockchain care este nucleul securității noastre.

Cuvinte cheie: *autonom, sisteme, computer, rețea, securitate, SDN, tehnologie, controler, topologie, algoritm.*

1. Introduction

The soaring number of applications for autonomous systems in different aspects like air, sea, and space is creating the need for new methodologies and architectures' technologies to consolidate the verification of system-level and system-of-systems level. The implementation is critical for backing up the cybersecurity standards and software infrastructure. The classical structure of networks has remained unchanged for a long time till the development of the new structure of software-defined network came up starting with its first attempt by Martin Casado which and that model was named Ethane [1]. Most security issues related to the heritage structure and hierarchy could be solved using the new paradigm of SDN but, due to the relative novelty of SDN methodology in managing the computer networks; many other new cyber-security challenges could emerge and with that comes the need to secure that environment to facilitate the transition from the classical network structure to the SDN structure and also there's a need to use risk assessment tools; both software and mathematical ones to determine the level of security risk that lies within a software-defined network environment since that it is useless or with less precision to use the common existing mathematical equations that were originally developed for assessing the security level in classical computer networks' environment.

There are many works and researches to use petri nets to study and analyze the security of computer networks in general like [2]; where they leveraged petri nets to model three defense scenarios the first one was with firewall only, the second was with firewall and Intrusion Detection System IDS and the third defense scenario was with firewall, IDS and a honeypot system. Also, many attempts to address the security issues in SDN in precise have been conducted including modeling the problems mathematically and graphically; the usage of petri nets was one of the main prominent methodologies used for that purpose [3] where they try in this work to model the SDN structure and model it under Denial of Service (DoS) attack but, not on the proposed topologies by this article. Also, in [4] the researchers tried to verify and give a general analysis for the security situation in the software-defined network environment that contains two switches and one controller; without the need to discuss the impact of a (DoS/DDoS) attack and again the modeled topology differs from those proposed in this article.

This article mainly discusses the topic of enhancing and ensuring the security of software-defined networks by suggesting several algorithms working together as a framework and by suggesting three different SDN controllers' topologies for the controllers of software-defined networks to overcome some cyber-security issues and to ensure the network reliability against various cyber-threats specifically, the threats of DoS/ DDoS [5].

We have discussed comparison. Also, this article presents a defense factor equation which is a new equation to assess the level of security risk in any SDN environment especially if it was based on any of the three presented structures in this research; since that most works haven't discuss that matter and even when they did, it wasn't based on the topologies proposed here and not discussing the previously mentioned attacks in precise as well and from that stems the importance and novelty of the research and this article that discusses it. And to discuss that specific matter, this article proposes the usage of petri nets modeling paradigm [9] to get simulation results to gain a better reading for the situation of the interaction between controllers and their networks and how would they react under the effect of DoS/DDoS attacks. In the section 2; we'll demonstrate briefly some of the main algorithms and methods proposed to form the framework that will be as a security solution each of them separately in other articles [6 - 8] but here we'll gather them all for the purposes of to patch different security breaches, in section 3 we'll talk about the main idea of the paper which is the Petri Nets modelling for the suggested SDN controllers' topologies to derive an equation that could be used as a mathematical tool to measure the defense capabilities of the software-defined networks that leverage our proposed models against different cyber-threats; especially the DoS/DDoS attacks. Least but not last there will be a conclusion for the whole paper in section 4.

2. Integrated algorithms into the proposed framework

There are few algorithms proposed in the research and we'll describe them briefly here in this paper. Those algorithms work together to form the whole framework that will be installed on top of the management layer or the application plane which.

Algorithms' suite integrated in HYDRA framework.

The framework will be installed upon the management plane; it is based on some algorithms and methodologies like the counter measurement attack against the perpetrator. Also it contains other technologies and algorithms as mentioned below.

Secured channel of VPN algorithm.

The secured channel provided by Virtual Private Network (VPN) technology has a great deal of security that it provides; its Internet Protocol Security (IPsec) algorithm can be leveraged to ensure the security in the communication channel between SDN controllers that is called the east-westbound API [10].

Cryptography of Double RSA algorithm.

The Rivest-Shamir-Adelman (RSA) algorithm is well known for its robustness in securing computer networks' communication till this day. This algorithm or methodology mainly consists of four stages, which are key generation, key distribution, encryption and decryption. The RSA algorithm contains two main keys' categories they are the public key and the private key. The stages of key generation are [11]:

1. The generation of two large prime numbers let us name them p and q.
2. The computation of n = p * q.
3. The computation of z = (p - 1) * (q - 1).
4. Computation of the mod z =v.
5. Selecting a prime number to z and let's name it as x.
6. Figure out the value of e. then we put: e*x = 1 mod z that means that: e*x =1*v.
7. Now let public key be (n, e)
8. Now let private key be (n, x).

It is proposed by this research to use it but in a full duplex-like way as shown in Figure 1.

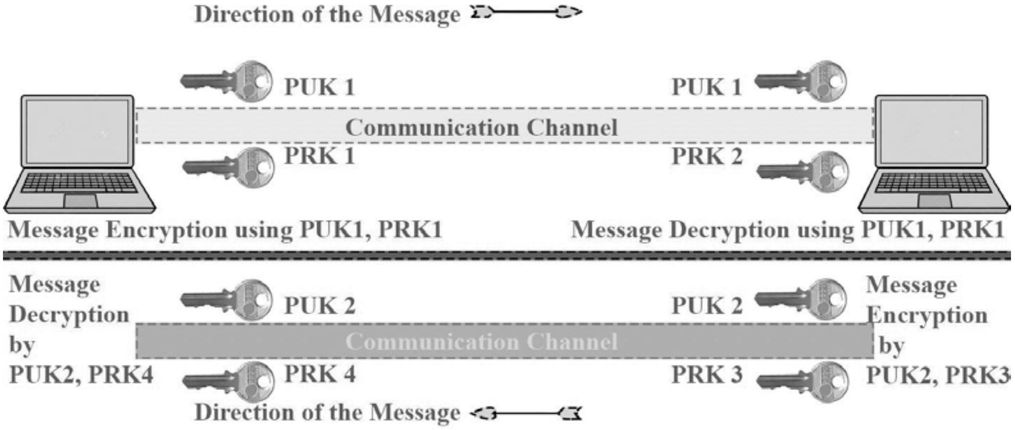

Direction of the Message
PUK 1
PUK 1
Communication Channel
PRK 1
PRK 2
Message Encryption using PUK1, PRK1
Message Decryption using PUK1, PRK1
Message Decryption by PUK2, PRK4
PUK 2
PUK 2
Message Encryption by PUK2, PRK3
Communication Channel
PRK 4
PRK 3
Direction of the Message

Figure 1. Double RSA.

Where there will be two public keys instead of one and four private keys instead of two creating to channels or tunnels for every node; where one of them will be used for sending information and the other one for reception.

Distributed ledger concept of Blockchain algorithm.

Blockchain is a promising technology and we have published an essay about that [12], it is a public ledger with a distributed feature that behaves like a log which could keep track of all transactions in a chronological way, it is secured using a mechanism of appropriate consensus and provides an immutable record [13]. Therefore, it is possible to say that the blockchain could be considered as a decentralized ledger for all peer-to-peer network's transactions. With blockchain, participants could conduct transactions' confirmation without any need for a controlling central authority. Potential usage applications could be trade contracts, voting, fund transfers, etc… [14]. The blockchain technology could be exploited for securing network configuration updates exchanged between multiple controllers in SDN structure, but in a different way than it is used in the Marconi protocol [15].

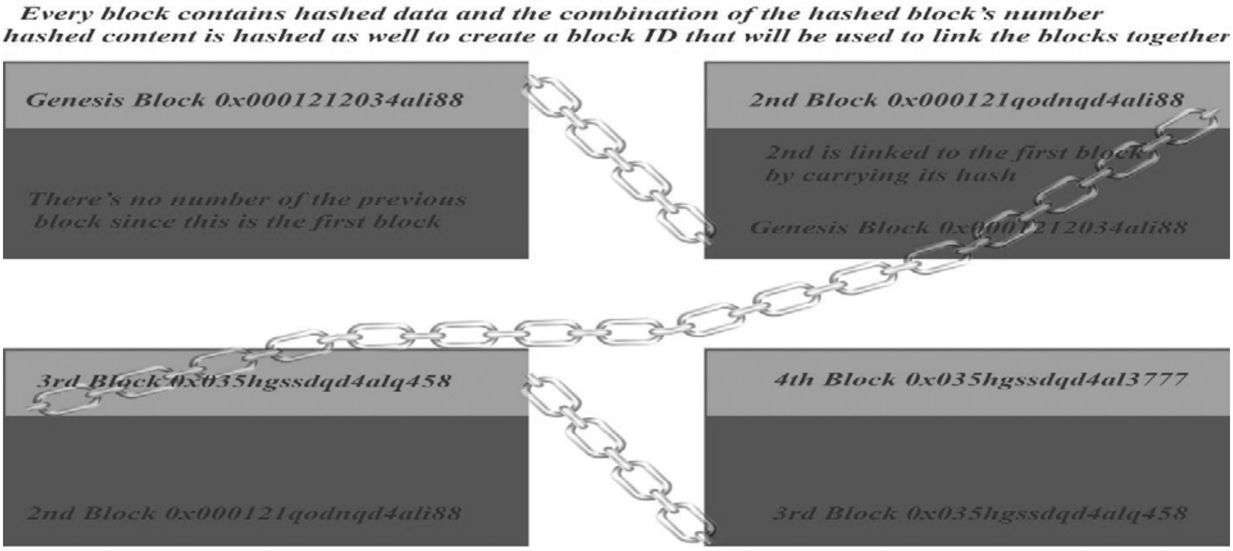

Every block contains hashed data and the combination of the hashed block's number
hashed content is hashed as well to create a block ID that will be used to link the blocks together
Genesis Block 0x0001212034ali88
There's no number of the previous block since this is the first block
2nd Block 0x000121qodnqd4ali88
2nd is linked to the first block by carrying its hash
Genesis Block 0x0001212034ali88
3rd Block 0x035hgssdqd4alq458
2nd Block 0x000121qodnqd4ali88
4th Block 0x035hgssdqd4al3777
3rd Block 0x035hgssdqd4alq458

Figure 2. Blockchain.

3. Modelling the proposed topologies with petri nets

In the beginning, it is important to give a simple description of the proposed topologies for software-defined networks' environment, that could be a better solution to deal with DoS/DDoS attacks.

Carl Adam Petri was accredited for the invention of Petri nets system and its aim is for describing chemical processes. The Petri Nets is a place/transition (PT) system; and it is a description and a modelling approach for distributed systems. Moreover, it has the definition of a dynamic system of discrete events. Petri Nets are directed bipartite graphs, therefore they're formed of two types of nodes. The two main nodes are places drawn as circles and transitions as bars. Those directed arcs define the direction of procedure and that would figure out which places are pre conditions and which are post conditions. The Petri Net presents a notation of a graphical nature that could be leveraged for diverse purposes and procedures with concurrent implementation. Also, this procedure presents a mathematical definition [3].

There are different types of simulation software that use petri net models so, it is possible to use one of the available choices we have; our choice here is the software named Platform

Independent Petri Nets Editor (PIPE). One of its main modules that we'll use for our own research is the Generalized Stochastic Petri Nets (GSPN) module that mainly focuses on what we exactly require and that's the amount of tokens that occupy the places which in turn represent the controllers placed in the SDN environment. The tokens here represent the updates or requests to or from the SDN controllers.

Now we are going to demonstrate how the three proposed topologies of controllers work , intercommunicate, and react towards each other. Also, we'll show how they defend themselves to deter different attacks like DoS/DDoS ones by modelling their behavior with petri nets system that uses Markov chains for modeling the behavior of the controllers in each topology. We will use the PIPE software to implement our topologies in petri nets, especially the module of GSPN to gain numerical results. The (GSPN) module, which stands for Generalized Stochastic Petri Nets, is defined as a 6-tuple (P, T, F, W, M_0, λ) [3] such that:

1. P= {P_1, P_2... P_m} describes the places' set n ⩾ 0 which are of finite nature.
2. While T= $T_1 \cap T_2$, T1= {t_1, t_2... t_m} describes finite timed transitions' group, and each one of those transitions has a delay time of a random nature the delay time is set within the period of enabling or activating and firing. In addition, T2= {t_{m+1}, t_{m+2}... t_n} is the finite immediate transitions' group or set, these transitions can be randomly fired and the delay value is set to zero.
3. Also, F⊆ (PxT) ∩ (TxP) is arcs' set; and, there are the inhibitor arcs which can only form places to transitions and disenable the activated conditions.
4. W represents the arcs' weight function: F →{1, 2, 3...}.
5. Also, M_0: P → {0, 1, 2, 3 ...} represents the initial marking, where (PxT) = Ø ∩ (TxP) = Ø [3].
6. λ= {λ_1, λ_2, λ_3 ... λ_n} represents firing rates' set that represent or describe the timed transitions. Each rate is the average firing times of transition in unit of time [3].

There three topologies presented in our research plus a fourth which is the single-controller Ordinary Topology. We'll compare the three topologies with the ordinary topology to depict the enhancements they have.

Serial topology

As shown in the figure below, this topology consists of three controllers interacting with each other and communicating as one main controller and two controllers as backup; we should mention that they could be software-oriented or hardware-oriented controlling entities.

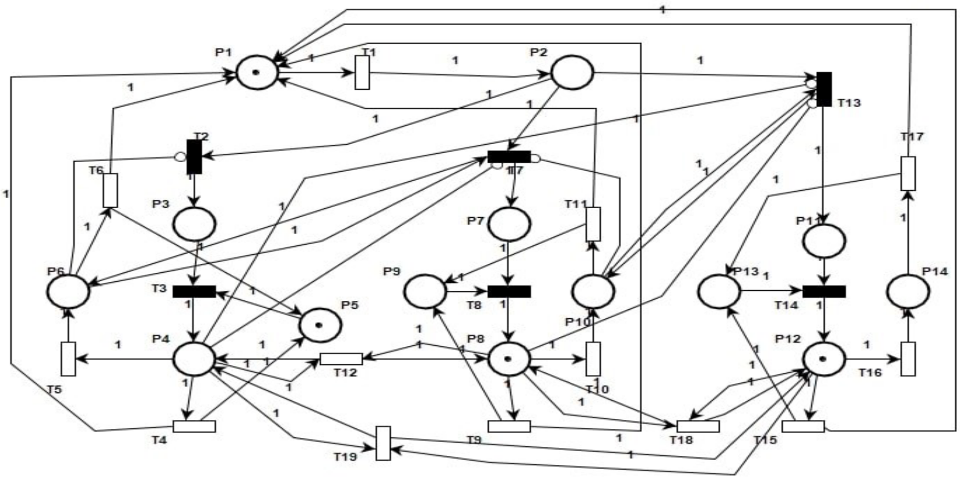


Figure 3. Proposed Serial Topology for Software-Defined Networks.

In this topology, the selected controller to work as the main controller is working normally and behaving as the decision-making node. This unit administrates the behavior of the network; communicating with the upper layer which is the management layer and with the lower layer which is the data layer and providing management for hosts' requests through the switches of the data plane.

The main controlling unit will send configuration updates and flow tables information of network every period of 10 seconds to the spare controllers to keep them up to date, to keep them as standby network controllers and to be able to continue administering and preserve the network information in case of a cyber-threat that could compromise the previous controlling node.

Again, after compromising the main node in the attack situation; the network management is going to be assigned to the standby controller that has the next number of priority, No. 2 controller for instance is going to be the one in charge in this situation. The next step will be sending a bot to infect the source IP of the attacker afterwards; this source IP of the attacker is going to be added to the blocked IPs in a single direction so the attacker can't send anything anymore just like the access control list (ACL) situation. The compromised controlling node is going to be isolated in the same time. The new elected controller will be the main one after it was the backup one previously. Now this controller administers the environment and exchanges network information with the other last remaining spare controller while the previously main controller is under maintenance. Table 1 provides Petri Nets places' description of the proposed Serial Topology model [6].

Table 1

Description of Places in Serial Topology for SDN

Place	Description
P1	Packets being processed by the main controlling node / the input place which sends data tokens
P2	Servers' selection
P3/P7/P11	Servers' allocation
P4/P8/P12	Active processing in the servers 1,2,3
P5/P9/P13	Free controllers 1,2,3
P6/P10/P14	Attack is active

The table 2 depicts the serial model transition' description [6].

Table 2

Description of Transitions in Serial Topology for SDN

Transition	Description
T1	Task generation i.e., processing of packets
T2/T7/T13	Selection of servers 1, 2, 3
T3/T8/T14	Allocation of servers
T4/T9/T15	Task processing
T5/T10/T16	Exiting the stage
T6/T11/T17	Restoring the controller
T12/T18/T19	Updating the information and going back to initial stage of controllers

Parallel topology.

This topology consists of three main controllers; which cooperate in managing the whole network as a one entity as if it was one controller. They update each other with the information they have about the contiguous network nodes each controller has as shown in figure 4.

As mentioned before in this topology we'll have 3 controllers as well but, the difference will be in their interaction with each other. The 3 controllers will be working as a whole entity like one brain of 3 parts where they work simultaneously to process switches' requests. They will all behave like the main controller. Each controller will serve switches and prioritize them based on the closest ones to it. Each one of them will send a broadcast update to other remaining servers/controllers of its configuration every 10 seconds as well.

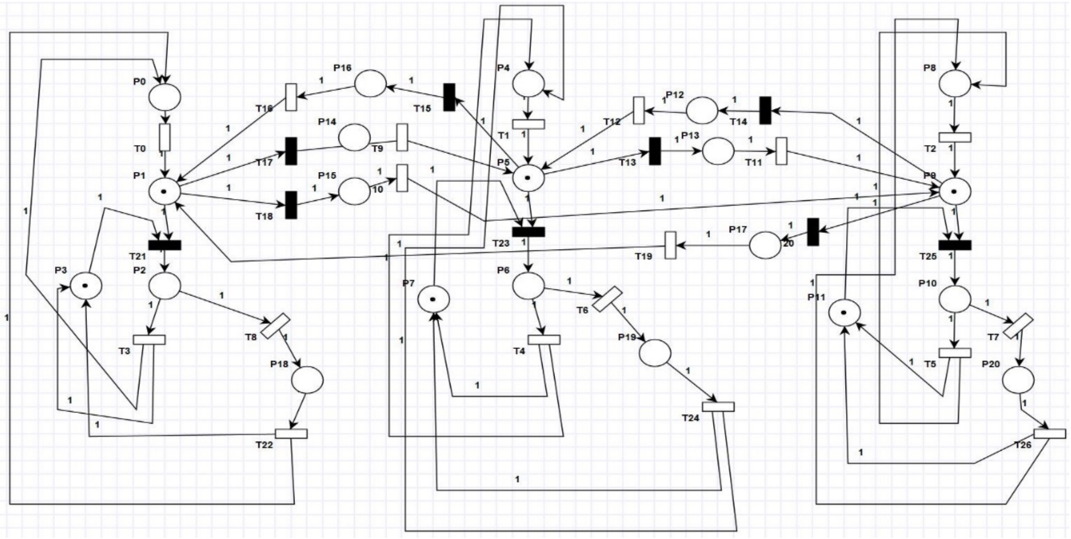


Figure 4. Proposed Parallel Topology for Software-Defined Networks.

Since all the three controllers represent a whole one main controller and the updates between the controlling entities will be every 10 seconds, that means that the configuration information will be merged together every 10 seconds, in other words the main triple-parts controller will update its general table of the whole network's status every 10 seconds. In case if there was a disruption on any one of the controllers, the other two controllers will fill the vacancy, left by the infected controller; and that's by serving the switches that were depending on the disrupted controller before it was disrupted. Of course; the switches will already be connected to all the three controllers so, in case of an attack; a bot that is already installed in the controllers; is going to be infiltrating the attacker's source IP. The attacked controller's IP will be isolated with the source IP of the attacker. Now all switches will be communicating with the controllers directly without any noticeable change, since that the remaining two controllers will add more space to deter Dos/DDoS attacks first and if this fails then the infected controller's IP will be isolated and the controllers continue their work like one controller of two main parts. The description of places is stated in the table 3.

Table 3

Description of Places in Parallel Topology of SDN

Place	Description
P1/P5/P9	Allocation of servers
P2/P6/P10	Server 1, 2, 3 Active processing

Continuation Table 3

P12-P17	Exchanging the information and configuration updates of the network
P3/P7/P11	Recovery and restoring the working state of the controller
P18/P19/P20	Attack on the server or controller
P0/P4/P8	Returning to the initial state

Description of transitions is in table 4.

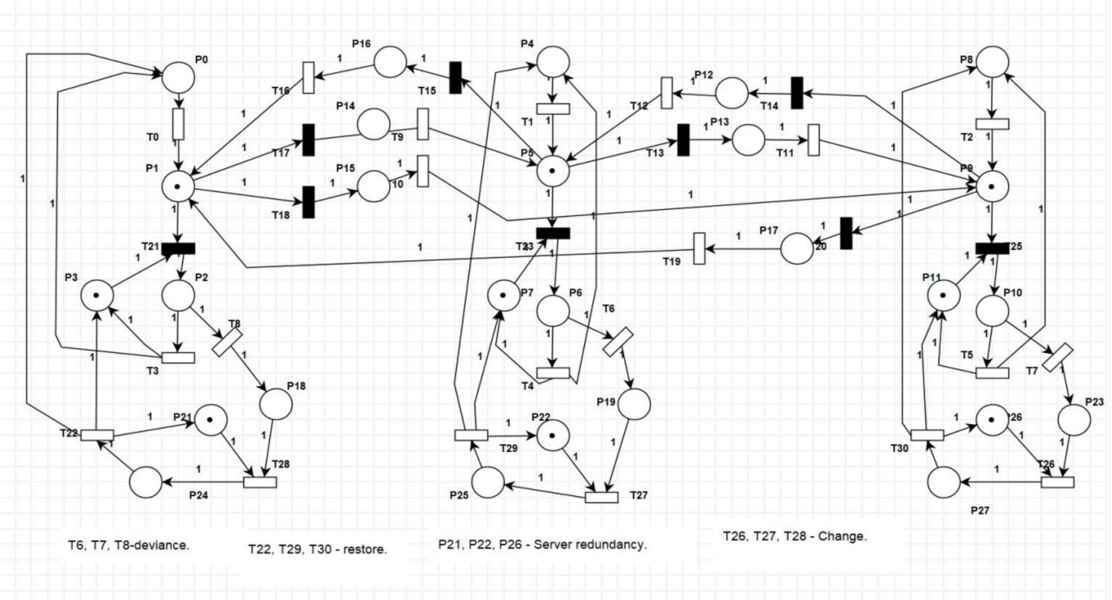
Table 4

Description of Transitions in Parallel Topology of SDN

Transition	Description
T21/T23/T25	Server 1, 2, 3 Active processing
T9-T20	Exchanging the updates of network configuration between servers
T3/T4/T5	Back to active processing/ processing the next request
T6/T7/T8	Attack on server/controller
T22/T24/T26	Recovery from the attack
T0/T1/T2	Transition back to initial state

Hybrid topology.

The Hybrid topology contains six controlling nodes; three primary controllers and three spare ones as backup. Every main controller has a backup controller that could the next dominant primary controller in the situation of a disruption or a threat on its own current main controlling entity. As shown in design 5.



T6, T7, T8-deviance.

T22, T29, T30 - restore.

P21, P22, P26 - Server redundancy.

T26, T27, T28 - Change.

Figure 5. Proposed Hybrid Topology for Software-Defined Networks.

In addition, this topology can be considered as a mixture of the Serial and Parallel topologies, therefore it was named as Hybrid; the Hybrid topology's structure consists of six controlling nodes. Three of those six are main controllers working together in a parallel way as whole one brain and in this aspect this topology resembles the parallel topology therefore, in this situation; the same rules of parallel topology can be applied [8]. The rest of the six controllers which are three will be considered as a backup for the main controlling nodes meaning, one backup controller for every main controller. The backup controllers will be connected to other network nodes as well and that's done via two ways:

- Being in connection with its main primary controlling node to take its place in the cyber-disruption or attack situation.
- Also, every single back controller will have a connection channel with everyone of the other backup or spare controlling nodes.

In critical situations of a disruption on any of the three main control layer nodes; then it will be blocked by isolation alongside with the perpetrator's source IP and it will be replaced with its substitute until the compromised previously main controller is back to work. Not to forget that prior to conducting this process; there will be an embedment of a bot within the source address of the attacker. The table 5 describes the places of this diagram [8].

Table 5

Description of Places in Hybrid Topology of SDN

Place	Description
P1/P5/P9	Allocation of servers
P21/P22/P26	Extra servers / backup servers
P2/P6/P10	Active processing
P18/P19/P23	Server/controller under attack
P24/P25/P27	Recovery of server/controller
P3/P7/P11	Processing next request
P0/P4/P8	Returning to initial state
P12-P17	Exchanging the information across the network

The table 6 describes the transitions of the hybrid topology in the model of Petri Nets [8].

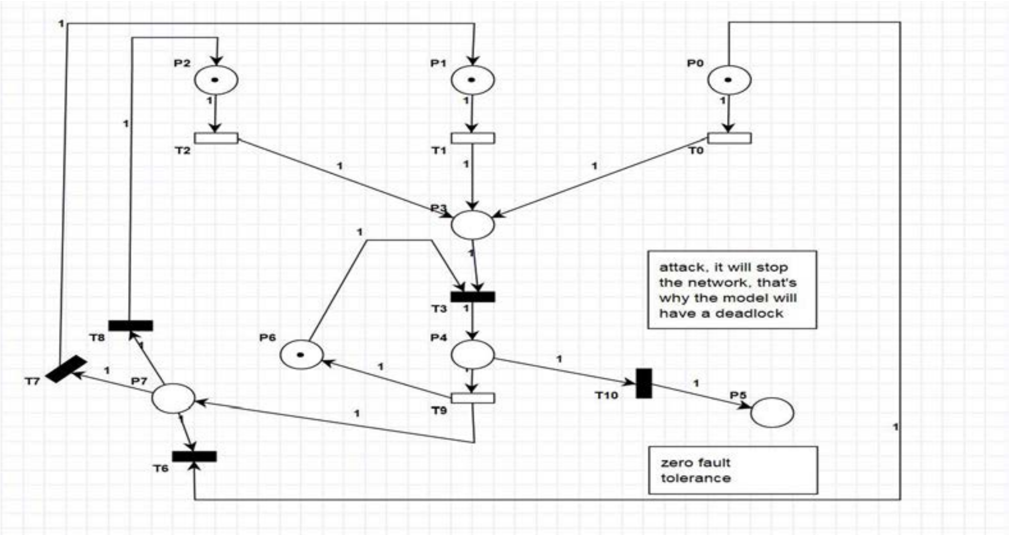
Table 6

Description of Transitions in Hybrid Topology of SDN

Transition	Description
T0/T1/T2	Transition from initial state to active processing
T21/T23/T25	Active processing
T3/T4/T5	Processing next request
T6/T7/T8	Deviance or attack state
T22/T29/T30	Transitioning to backup /restoring/ back to initial state
T26/T27/T28	Transformation and change
T9-T20	Sharing and updating the network configuration between servers

Ordinary topology

This modeling is conducted for representing the usual single-controller topology that has the name; the Ordinary topology in our research which has one controller and for describing its weakness points.



attack, it will stop the network, that's why the model will have a deadlock

zero fault tolerance

Figure 6. Proposed Ordinary Topology for Software-Defined Networks.

This topology describes a simple single-controller environment of Software-defined network. It is being mentioned and modelled here for comparative analysis purposes to depict the reliability of the presented framework alongside its suggested three environments [8]. This diagram describes the vulnerability of single-controller SDN environment and how it is attack-prone due to the single point of failure SPOF issue; that needs to be patched. This network is managed via a single controlling entity that deals with switches' requests as usual until an attack occurs. In the attack situation any kind of disruption, the aforementioned model proves that a simple attack focused on that only controller could jeopardize the whole network. That shows that this single-controller environment has a fault tolerance of zero value. The Table 7 depicts the single-controller diagram's places [6].

Table 7

Places description in Ordinary Topology of SDN

Place	Description
P0/P1/P2	Selection of switches
P3	Main controlling node
P4	Active processing
P6	Next request Processing / return to the initial state
P7	Exchanging requests
P5	Compromise attempt on the controller

The table 8 describes transitions of Ordinary topology in a Petri Nets model [6].

Table 8

Description of Transitions in Ordinary Topology of SDN

Transition	Description
T0/T1/T2	Providing the controller with requests
T3	Active processing
T9	Initial state/ dealing with switches
T6/T7/T8	Selection of switches
T10	Attack attempt

By conducting a simulation in the PIPE software using the GSPN module on the above models; it is possible to gain the tokens' average number in that lies in every controller to derive a mathematical relationship between the used topology's type and the average number of tokens that represent that requests dealt with by each place that represents a controller. In addition, it is obvious that the bigger the number of tokens or requests; the less the defense ability the controllers will have so that means that defense factor value has an opposite proportion to the number of tokens. One more thing to notice that the proposed mathematical relationship has to show the same results gained by the PIPE software which clearly state that the parallel topology is the best amongst all of the other topologies since it has the least average number of token/requests per unit of time and that what we have to model mathematically. Table 9 describes the tokens' average number in places/controllers in the models of the Petri Nets topologies [6], [8].

Table 9

Average Tokens' number in SDN Controllers represented by PN Places in GSPN Module

Algorithms **Places**	**Serial Topology**	**Parallel Topology**	**Hybrid Topology**	**Ordinary Topology**
P3	0.16337			1.99975≈2
P7	0.06867			
P11	0.13233			
P1		0	0	
P5		0	0	
P9		0	0	
P21			0.9037	
P22			0.90368	
P26			0.90352	

The needed and created mathematical relationship is a relationship that could achieve the same results, as it will be shown afterwards. This relationship proved that based on the gained numerical results; it could be inferred that all the newly proposed three topologies work and behave better than the single-controller ordinary topology. Especially, the Parallel topology that is the ideal one theoretically and, from the mathematical relationship that was suggested in this research; we could acquire the same results and implications that match the acquired numerical results from the aforementioned simulation.

So, based on the simulation results, we propose an equation that represents the relationship between the number of controllers, type of interaction between them and the number of requests directed from/to them. The formula is derived to create a security factor that could be used a standard for determining the security intensity of any software-defined network that is leveraging any these three previously mentioned topologies [8]:

$$\Delta = \sum_{i=1}^{i=n} k_i \cdot \frac{1}{\sum_{z=0}^{z=\infty} z_{k_i}}$$

Where K is the places number or amount, which describe the work of the controllers in every model, Ki= (K1, K2… Kn) and Z represents the number value of tokens in those places K of those models. In addition, Zki = (0… ∞). Of course, the Higher the value of the Defense Factor, the better it is. By applying the gained simulation results, it is possible to find that according to the law the best topology would be the 2nd one or the parallel topology and that matches the simulation results shown in the table, that shows that the controllers in the 2nd topology will be less occupied with requests than other topologies which means that they will be more available hence; have more reliability against Dos/DDoS attacks.

The article's scientific value focuses mainly on figuring a method to find a measurement for the security proportion or intensity of any network environment that leverages any one of the aforementioned topologies. The article here depicted those topologies briefly and compared between them and the Ordinary topology.

The proposed formula's reliability and correctness is inferred from its matching to the simulation gained results. Because both the acquired results and the formula's implementation prove and assure that all the suggested SDN controllers' topologies work in a better way than the ordinary one. In addition, those topologies' controllers are freer along the average time of processing meaning that; they are less busy than the controller of the ordinary topology which means that those controllers be more reliable and more deterring to attacks like DoS/DDoS which focus on submerging the controllers with pseudo requests.

4. Conclusions

We have presented in this article an idea about assuring the security of networks by assuring the security of software-defined networks hence, making it easier for classical structure networks to change to the SDN paradigm since it will be safer.

We discussed the main methods proposed to be integrated with the application layer as an application that will be a full framework to assure the security of SDN.

Alongside the proposed algorithms, here we explained briefly about the proposed working topologies in our research, that categorizes three main working topologies for software-defined networks' controllers.

Here we mainly focus on modelling the proposed topologies using Petri Nets system, to conduct a simulation on those modelled designs using PIPE software and to gain results that could show the best topology to be used with the emptiest space in its controllers; meaning the topology with the least vulnerable controllers to DoS/DDoS attacks. Based on the gained results we have created a mathematical relationship that will simulate the pattern of the gained results and their implications. This formula can be used as a theoretical mathematical instrument to figure out the cyber-threat level imposed on a network that leverages the SDN structure.

When we apply the acquired results from the simulation on this presented defense factor formula, the same implications could be gained. From the numerical simulation results, it is possible to conclude the following:

All the three proposed topologies are better than the usual one-controller SDN environment, which is the ordinary topology.

The parallel topology is the best of all four topologies since its controllers are emptier most of the average processing time.

And using the defense factor formula; it is possible to get a match for the gained numerical results which proves its feasibility.

The article's scientific value focuses mainly on figuring a method to find a measurement for the security deterrence level of any SDN that leverages any one of the aforementioned environments. This article depicted those topologies briefly and compared between them and the Ordinary topology.

The proposed formula's reliability and correctness is inferred from its matching to the simulation gained results. Because both the acquired results and the formula's implementation prove and assure that all the suggested SDN controllers' topologies work in a better way than the ordinary one. In addition, those topologies' controllers are freer along the average time of processing meaning that; they are less busy than the controller of the ordinary topology which means that those controllers be more reliable and more deterring against cyber-attacks which focus on submerging the controllers with pseudo requests.

References

1. Casado M. et. al., Ethane: Taking Control of the Enterprise. In: *ACM SIGCOMM Computer Communication Review*, 2007, 37(4), pp.1-12.
2. Shi L., LI Y., Feng H. Performance Analysis of Honeypot with Petri Nets. In: *Information*, 2018, *9*(10), pp. 1-19.
3. Almutairi L., Shetty S. Generalized Stochastic Petri Net Model Based Security Risk Assessment of Software Defined Networks. In: *Military Communications* (MILCOM), 2017, pp. 545 - 550.
4. Zhou Y., Bai Y., Zhao P. A Petri Net-based Method to Verify the Security of the SDN. In: *IOP Conference Series: Earth and Environmental Science*, 2018, 234(1), pp. 1 - 6.
5. Prasad K., Reddy A., Rao K. DoS and DDoS Attacks: Defense, Detection and Trace Back Mechanisms – A Survey. In: *Global journal of computer science and technology: E Network, Web and Security*, Issue 7, 2014, JNTUH University, India. Volume 14, 18-pages.
6. Ameen A., Guţuleac E. Petri Nets Modeling for SDN topologies. In: *Journal of Engineering Science*, 2021, Technical University of Moldova, 28(2), pp.79-90.
7. Ameen A. Modelling sdn's parallel topology using petri nets. In: *Journal of Engineering Science*, (to be published).
8. Ameen A. Modelling proposed hybrid software-defined network controllers' topology using petri nets system. In: *Studia Universitatis Moldaviae*, 2021, Moldova State University, 2(142), pp.40-50.
9. Petri Net. [Online]. [accessed 05.01.2020]. Available: https://en.wikipedia.org/wiki/Petri_net.
10. Jyothi K., Dr. Reddy B., Study on Virtual Private Network (VPN), VPN's Protocols and Security. In: *International Journal of Scientific Research in Computer Science*, 2018, Sreenidhi Institute of Science and Technology, 3(5), pp. 919-932.
11. Ukwuoma H., Hammawa M., optimized key generation for RSA encryption: thesis. Ahmadu Bello University Zaria-Nigeria, 2015.
12. Ameen A. leveraging blockchain technology to assure security of SDN. In: *Journal of Engineering Science*, Proceedings of International conference on Electronics Communications and computing, October 2019, pages 128-139.
13. Puthal D., Movants S. P. *Everything you wanted to know about Blockchain: its Promise, Components, Processes and Problems*, Australia-USA, 2018.
14. Herweijer C., Waughray D., Warren S. Building Block (chain)s for a better planet, world economic forum, 2018.
15. Marconi foundation. (2018). Marconi Protocol: white paper. Retrieved 25.08.2019, from Marconi: https://docsend.com/view/5zragmb.