STUDIA UNIVERSITATIS MOLDAVIAE, 2021, nr.2(142)
Seria "Științe exacte și economice" ISSN 1857-2073 ISSN online 2345-1033 p.40-50

CZU: 004.72 + 519.711 **DOI: http://doi.org/10.5281/zenodo.5094689**

MODELLING PROPOSED HYBRID SOFTWARE-DEFINED NETWORK CONTROLLERS' TOPOLOGY BY USING PETRI NETS SYSTEM

Ali AMEEN

Technical University of Moldova

There's a need to secure the paradigm of software-defined networks for many reasons, and one of the methods proposed in our research is to use Petri Nets system to model some proposed working topologies for the SDN controllers; those topologies were also proposed in our research to secure some aspects of the SDN paradigm. After modelling the suggested SDN controller topologies, it is possible to imply that modelling in the PIPE software to get simulation results. Those results could be used to derive a security equation to measure the security level of any network that's based on the SDN structure and leveraging any of the three proposed topologies in our research. In this paper we'll concentrate on modelling the third and last proposed topology which is the Hybrid topology and compare it with the single-controller already existing topology that we named the Ordinary topology. The comparison will show the feasibility of the Hybrid topology and its advantageous effect over the Ordinary topology.

Keywords: *Blockchain, GSPN, hydra, Hybrid topology, Petri Nets, RSA, virtual private network.*

MODELAREA PROPUNERII TOPOLOGIEI CONTROLERELOR DE REȚELE DEFINITE DE SOFTWARE HIBRID UTILIZÂND SISTEMUL PETRI NETS

Este necesar să se asigure paradigma rețelelor definite de software din mai multe motive și unul dintre metodele propuse în cercetarea noastră este utilizarea sistemului Petri Nets pentru a modela unele topologii de lucru propuse pentru controlerele SDN. Aceste topologii de asemenea au fost propuse în cercetarea noastră pentru a asigura unele aspecte ale paradigmei SDN. După modelare, este posibil să se presupună modelarea în software-ul PIPE pentru a obține rezultate de simulare. Aceste rezultate ar putea fi utilizate pentru a obține o ecuație de securitate pentru a măsura nivelul de securitate al oricărei rețele care se bazează pe structura SDN și care utilizează oricare dintre cele trei topologii propuse în cercetarea noastră. În această lucrare ne vom concentra pe modelarea celei de-a treia și ultimei topologii propuse, care este topologia hibridă, pe care o vom compara cu topologia cu un singur controler deja existentă, pe care am numit-o topologie obișnuită. Compararea va arăta fezabilitatea topologiei hibride și efectul său avantajos asupra topologiei obișnuite.

Cuvinte-cheie: *Blockchain, GSPN, hidra, topologie hibridă, rețele Petri, RSA, rețea privată virtuală.*

Introduction

The software-defined networking environment is a great way and a robust approach for solving common network issues, since it's a new way of managing computer networks but, as mentioned before in other articles and papers we have published [1-3]; it does raise some other cyber challenges and gaps in the same time therefore; it is needed to solve and patch those problems and gaps. In previous article [4] we have mentioned the advantages of software-defined Networks and we have mentioned the main points and issues that our research is targeting to solve in the SDN paradigm and in a brief, they are:

- **Centralisation:** Despite that centralisation of SDN architecture is one of the main positive features of SDN and an advantage in SDN over the classical architecture on one hand, it represents a potential threat itself in the same time by creating a single point of failure on the other hand.
- **East-westbound API:** In multi-controller topologies there is a channel that connects between every two controllers and that channel is an application programming interface API called east-westbound API [5] since the connection between the higher planes and the lower planes like control and data planes is referenced by the directions north and south. There is not much concentration on them, and it could be vulnerable to some cyber-attacks like MITM [6], DoS or DDoS [7] types of attacks which have a destructive effect [8].
- **Security level assessment and defence ability measurement:** most works try to do some modelling for computer networks and measure their security level based on some existing general laws of risk assessment or try to conduct some specific mathematical analysis for that particular instance of network. To the best of our knowledge, there's no fixed solid security level assessment methodology for software-defined networks due to many reasons like: their dynamic and ever-evolving properties, various topologies, different numbers of controllers used, etc.

This article will mainly concentrate on solving the first and third issues which are the centralisation point and the mathematical tool needed for measuring the security integrity of a computer network that is based on the SDN structure, especially if it was leveraging any of the proposed controllers' topologies. In this article, we'll mainly focus on the last proposed topology of the three proposed ones, which is the Hybrid topology and we'll give an explanation for it against the single-controller topology that we named in our research as the Ordinary topology then, we'll try to model them by using the Petri Nets system and conduct a simulation by using a Petri Nets-based software called Platform Independent Petri Nets Editor (PIPE). In the simulation we'll produce numerical results that could be used for comparison between the two previously mentioned topologies. After that, there will be an extrapolation based on those numerical results alongside the previously gained results for the other topologies and that extrapolation will be used to derive a security level assessment equation that we call the defence factor formula that could be used to determine the security condition level in the software-defined networks that leverage any of the proposed topologies. Based on that formula, we can show a simple comparison between the Hybrid topology and the Ordinary one and depict the enhancements made in the Hybrid topology by showing its reliability over the Ordinary topology.

The main purpose of this paper is to assure the security of the software-defined network paradigm by leveraging a mathematical formula to determine the cyber-threat level posed, to measure the defence and deterring ability of the network against cyber-attacks, especially Denial of Service/Distributed Denial of Service DoS/DDoS attacks [9]. The formula proposed in this research is derived based on the Petri Nets system that is simulated by using the PIPE software, meaning that the proposed topologies were modelled using the Petri Nets system and the Petri Nets modelling was simulated using the PIPE software to get numerical simulation results. Based on those results gained from the simulation, the mathematical relationship will be drawn, this relationship can be implemented on the same proposed topologies to depict a comparison of behaviour between those topologies to determine their security level and figure out the best topology and the most capable of deterring cyber-threats. The scientific value of this paper revolves about determining a tool for measuring the risk level/ security level of any network that is based on the SDN paradigm and specifically if it was comprising or consisting of any of the three proposed topologies in this research. This paper depicts one of those three proposed SDN controllers' topologies which is the Hybrid topology and gives a comparison between it on one hand and the single-controller topology that we call in our research the Ordinary topology. The robustness and correctness of the proposed equation is derived from its matching to the numerical results gained from simulation of the PIPE software, since both simulation results and the implementation of the formula emphasize that the Hybrid topology is more reliable and its controllers are more free along most of the average processing time hence, they're emptier than the single controller of an ordinary single-controller topology and that means that their defence capability is higher against DoS/DDoS attacks that mainly aim to submerge servers with fake requests.

The proposed solutions for the previous issues could include a package of both algorithms and SDN controllers' topologies combined and optionally integrated together as a full framework that could be incorporated with the SDN environment to enhance it and patch up the previously mentioned issues.

First, we need here to describe the proposed suite of methods and algorithms briefly, those methodologies are optional to be integrated with the three proposed topologies or with any other SDN controllers' topology and after that the topologies will be discussed briefly as well focusing mainly on the proposed Hybrid topology and the derived defence factor formula.

In section 2, we'll talk about the tools that are used for the modelling and simulation; next, in section 3, there will be a brief explanation of the main suggested algorithms and after that, in section 4 comes the main topic of this article which is the proposed topologies, where we mainly concentrate on the Hybrid topology that we propose in comparison with the single-controller Ordinary topology. In section 5, there will be a talk about the Petri Nets modelling of both the proposed Hybrid and Ordinary topologies. After that, in section 6, there will be the formulation of the defence factor formula and explanation of its basis. Last but not least comes the section of conclusions which is number 7.

1. Materials and laboratory tools

There are many kinds of simulation software that could be used for modelling Petri Nets system so, in our research the software of choice will be the platform independent petri nets editor (PIPE) software and we'll leverage one of its main modules for simulation and that is the Generalized Stochastic Petri Nets (GSPN)

module which mainly focuses on what is needed exactly here for this research and that is reviewing the number of tokens occupying the places that represent the controllers of SDN. GSPN stands for Generalized Stochastic Petri Nets which is a 6-tuple (P, T, F, W, M_0, λ) module where [4]:

1. P= {P_1, P_2… P_m} is a finite set of places, n ≥ 0.
2. T= $T_1 \cap T_2$, T1= {t_1, t_2… t_m} is a finite set of timed transitions, and each of these transitions is associated with a random delay time between enabling and firing. And T2= {t_{m+1}, t_{m+2}, …, t_n} is a finite set of immediate transitions, which can be fired randomly, and the delay is zero.
3. F⊆ (PxT) ∩ (TxP) is a set of arcs; also, there exit inhibitor arcs that can also form places to transitions and make the enable conditions to be disenabled.
4. W is a weight function of arcs: F→{1,2,3, ….}.
5. M_0: P → {0, 1, 2, 3, …}is the initial marking, where (PxT)= *Ø* ∩ (TxP)= *Ø*.
6. λ= {λ_1, λ_2, λ_3, … λ_n} is a set of firing rates corresponding to the timed transitions. Each rate is the average firing times of transition in unit of time [10].

By using the Petri Nets system that lies in the GSPN module of the PIPE software, it's possible to design a theoretical environment for the proposed topologies to acquire simulation results and based on them derive a mathematical relationship that could be leveraged to measure the security level/ threat level of networks which are based on the SDN paradigm and comprise one or any of the proposed topologies in this research and to see their defence capability against cyber-threats like DoS/DDoS attacks. The designs were based on the proposed topologies and their correctness is derived from:

- Their feasible and successful work.
- There are no deadlocks, which proves that the design is working properly.
- The software itself doesn't give results if the topology doesn't work but, the PIPE software generated results in our case.

This paper will demonstrate how one of the suggested topologies of controllers which is the Hybrid topology behaves in a work environment and how its controllers interact with each other and against a DoS/DDoS attacks. Then, this paper will draw a comparison between the suggested topology and its numerical results on the one hand and the gained results of the already existing simple ordinary one-controller topology on the other hand. The received numerical results by the PIPE software are automatically calculated by the software itself based on the designs that we already provided and explained their correctness. And on the basis of those gained numerical results, we derived the mathematical equation of defence factor. This equation or relationship could fulfil the same implementation as we'll see later, which proves that based on the numerical results it is possible to conclude that the Hybrid topology is better than the ordinary topology and, using the relationship that we suggest, it is possible to gain the same implications that match the gained numerical results.

The basics of the proposed defence factor formula are the gained results from software simulation, the relationship between them and the difference between them, based on those results; a probabilistic equation was derived to simulate and match the acquired numerical results. The equation gives a theoretical probabilistic measurement for the feasibility of each topology leveraged by the software-defined network environment.

2. Suggested Algorithms

In our research we elaborated and proposed some methods and algorithms to be integrated together in a whole framework to solve the problems noticed in the research and they're briefly discussed here in this paper:

- Hydra: this paper shows what we designed and proposed the framework that contains the algorithms and that will be mentioned later here. Hydra is formed of some techniques integrated together with those methods; techniques like counterattack measurements to counter attack the Denial of Service/ Distributed Denial of Service (DoS/DDoS) attacks [9], by installing botnets [11,12] into software-defined network computers that are connected with the SDN controller that has the Hydra software installed to make them as potential zombie guards to attack the attacker's source IP.
- VPN: one of the main methodologies or techniques used in the Hydra is the virtual private network (VPN) which has a well-known reliability in securing communication channels through its Internet Protocol Security (IPsec) technique. It can create secure communication virtual channels between network nodes and since it is widely used in different networks then, it is possible to add it to the proposed framework. The VPN is needed to specify a certainty that the confidentiality of sensitive data can be kept transmitted on the network a Local Area Network (LAN) or workable so that only authorised users are able to access sensitive data [13]. Basically, it's possible to connect two controllers via the VPN secure channel even if they were in the same location and exchange securely the information between each other.

- **RSA:** RSA (Rivest–Shamir–Adleman) is one of the first public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and it is different from the decryption key which is kept secret (private). In RSA, this asymmetry is based on the practical difficulty of the factorization of the product of two large prime numbers, the "factoring problem". The acronym RSA is made of the initial letters of the surnames of Ron Rivest, Adi Shamir, and Leonard Adleman, who first publicly described the algorithm in 1977 [4]. Clifford Cocks, an English mathematician working for the British intelligence agency Government Communications Headquarters (GCHQ), had developed an equivalent system in 1973, but this was not declassified until 1997 [14]. This algorithm is already used in almost every network communication these days but it is included by us here in this framework in a modified approach and that's done by doubling the channel of cryptographic communication; which means that instead of using one public key and two private keys for every encryption-decryption process, the proposed Hydra framework will use two public keys and four private keys, hence, every node will have a channel for sending encrypted information and a channel for receiving information meaning that there will be two channels of encrypted communications.
- **Blockchain:** which is also incorporated with the aforementioned framework but in a different way [4]. As it is known, blockchain's best and biggest participation is in cryptocurrencies like bitcoin but, it is also used in some other fields and it is already used in the Marconi protocol [15], but here we have provided a different usage design for blockchain to ensure the security of configuration updates exchanged between the multiple controllers that are configured and distributed in the proposed three topologies in our research.

3. Proposed Topologies

We have suggested in this research different topologies to overcome the centralisation issue which is already an advantage over the classical structure of networks since it is giving the software-defined network's structure the capability to manage the whole network and enforce policy in an agile approach but, at the same time it could be used as a vulnerability by being a single point of failure (SPOF) in case of an attack on that point which is the single controller that manages the whole structure of the network. These suggested topologies could use the proposed Hydra framework optionally and whether those topologies activate the framework or not, they already can help to overcome the centralisation issue by themselves. Those topologies differ in how many controllers they consist of and the type of interaction between them. In this paper we'll focus on the Hybrid topology which is the last proposed topology of the three ones that we suggested in our research and despite that it could have already been in use by some researches but here the difference will be in the type of interaction between the controllers themselves.

- **Hybrid Topology**

As we can see from figure 1 below, this topology combines features of both the previous topologies where we have here 6 controllers. There will be 3 main controllers that work like one controller simultaneously and in a parallel way and each one will have a backup controller just in case if it's down then, the backup will take control instead of the infected one. The only update will be between each main controller and its backup one, and it will be every 10 seconds as well. Every backup controller will be connected to other backup controllers alongside with switches in the network and its own main controller that it assists as well. The priority numbers will be between every main controller and its backup one only.

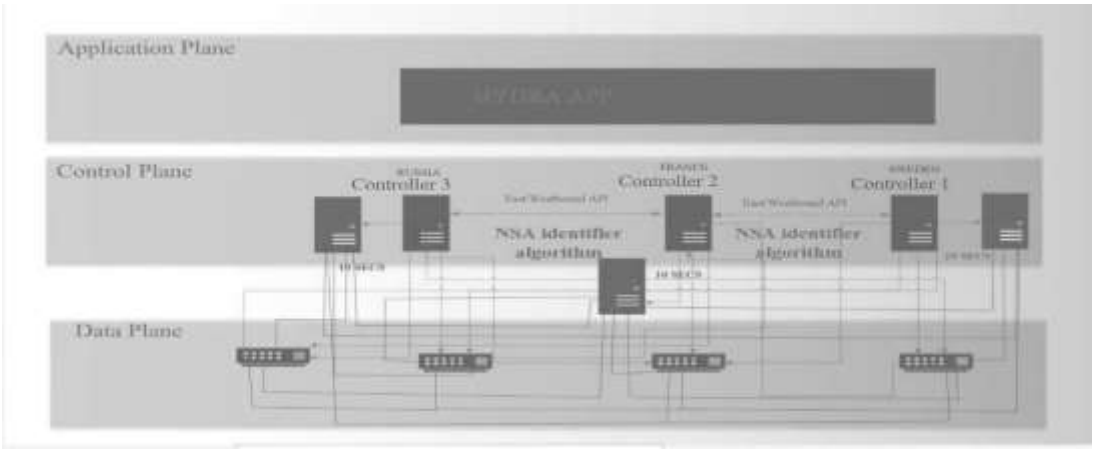

Fig.1. Hybrid Topology.

The fourth topology which is the ordinary usual one containing a controller the behaves as the brain of the network; fails to achieve its role in the existence of these ever-evolving threats and this topology fails to implement the proposed algorithms and topologies as well, because they are meant for multiple controllers' topologies.

- **Ordinary Topology**

Here we have a basic topology of software-defined networks, where we have one controller that controls the whole network, it controls the switches and they control the rest of the network of course; here the controller will be serving the computers by serving the switches that transfer the requests of the computers.

But, here if we have too many computers requesting to be served or a DoS/DDoS attack on the controller and that attack was somehow able to disrupt the server/controller and we don't have a backup controller that works with our main controller hence, that will stop the controller with no substitute and it might jeopardize the whole network by stopping it or hacking into switches by controlling the controller itself or by giving the network commands to let unauthorized entities or devices to be connected to the network and that will mean the end with no ability to recover. Figure 2 below shows an example of the ordinary one-controller topology.

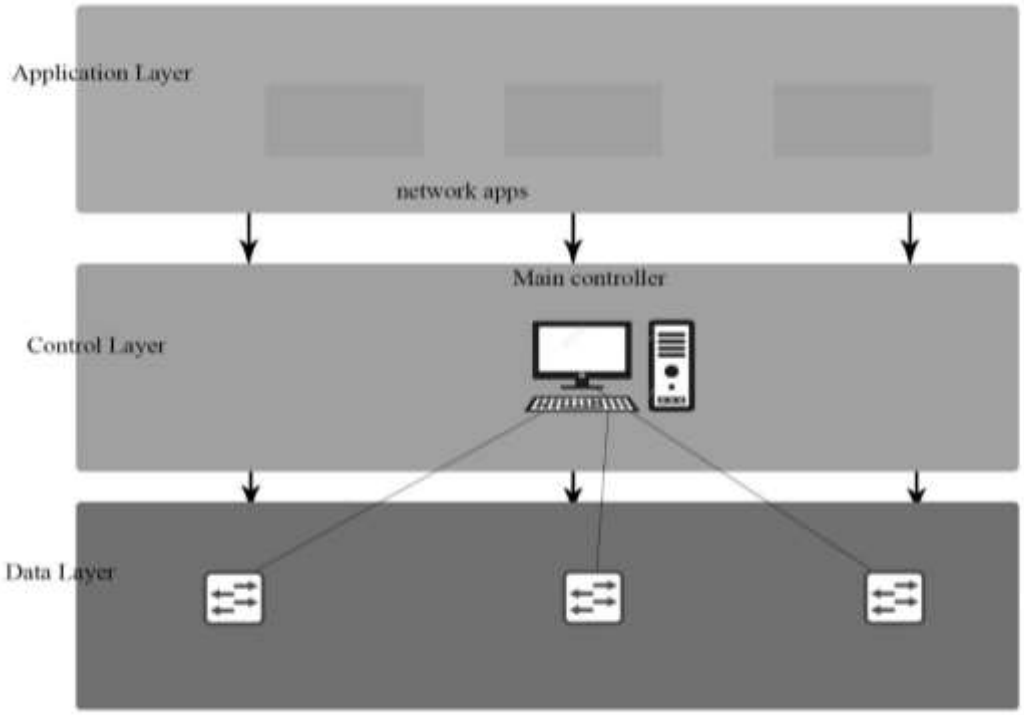

Fig.2. Ordinary Topology.

4. Petri Nets Modelling

Petri nets field was invented by Carl Adam Petri for the purpose of describing chemical processes. A Petri net is called a place/transition (PT) net as well. It is described as one of many available mathematical modelling techniques used for the purpose of modelling distributed systems. Also, it could be described as a discrete event dynamic system. The petri net is a directed bipartite graph, meaning that it contains mainly of two types of nodes which are places (i.e. conditions, represented by circles) and transitions (i.e. events that may occur, represented by bars). The directed arcs or arrows describe the direction of the procedure meaning which places are pre- or post-conditions for which transitions. Petri nets technique offers a graphical notation for stepwise procedures or processes that could include iteration, concurrent execution and/or choice. This technique has an exact mathematical definition [16]. We have used petri nets for modelling of the Hybrid topology, to have a better understanding of the topology's capabilities and to derive a formula from its behaviour that could be leveraged to measure the security level of a software-defined network especially if it was based on that topology.

- **Hybrid topology**

Here, the topology consists of 6 controllers; 3 main controllers and 3 backup ones, where each main controller has one backup controller to be used as the main controller in case of a disruption, termination or any kind of attack on that main controller. The figure 3 below shows the Hybrid topology modelling.

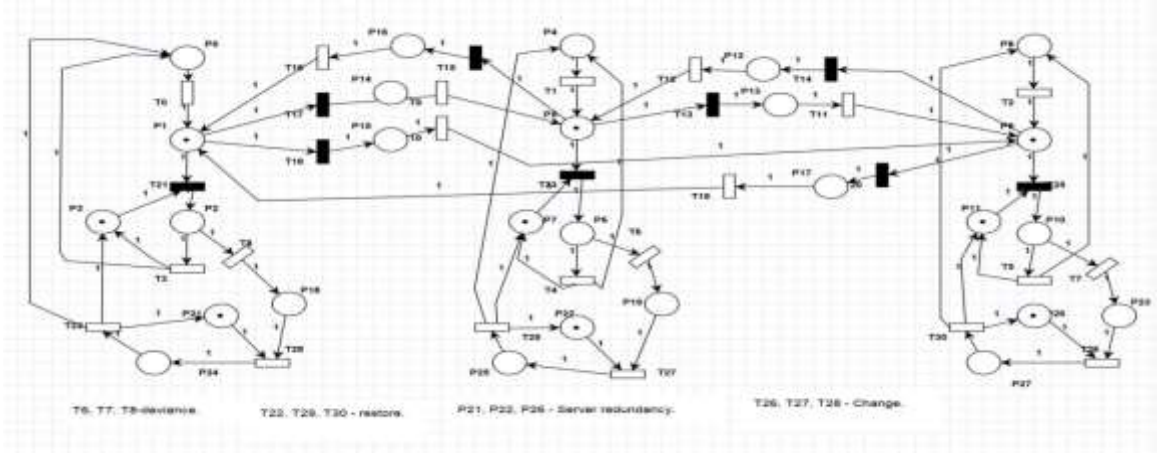

Fig.3. Hybrid topology modelling by using Petri Nets.

Description of the Hybrid topology scheme:

1. Also, as previously mentioned, this topology is a mix of both the previous topologies, hence comes the name Hybrid; the structure of the proposed formation of controllers will contain 6 controllers.
2. There will be 3 main controllers working in parallel as one integrated entity just like the parallel topology hence, in this case parallel topology rules apply here.
3. Every node or part of the triple main controller will have its own backup controller which will be also connected to the network through 2 ways:
    1) Connected to its main controller to replace it in case of an attack on its main controller.
    2) Connected to the other backup controllers.
4. In case of an attack on any main controller, it will be isolated alongside with the attacker's IP and it will be replaced with its substitute or backup controller till the maintenance of the infected controller finishes and of course, before embarking that procedure, a bot will be sent to the attacking source. Table 1 down below gives a description of the places of Petri nets diagram.

Table 1

Description of Places

Place	Description
P1/P5/P9	Servers' allocation
P21/P22/P26	Server redundancy/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	Back to initial state
P12-P17	Sharing the information and updating the network configuration

Table 2 contains a description of the transitions of the diagram of the Hybrid topology.

Table 2

Description of Transitions

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	Change
T9-T20	Sharing and updating the network configuration between servers

- **Ordinary Topology**

As shown in the figure 4, this modelling represents the usual ordinary topology with a single controller and shows its weakness points.

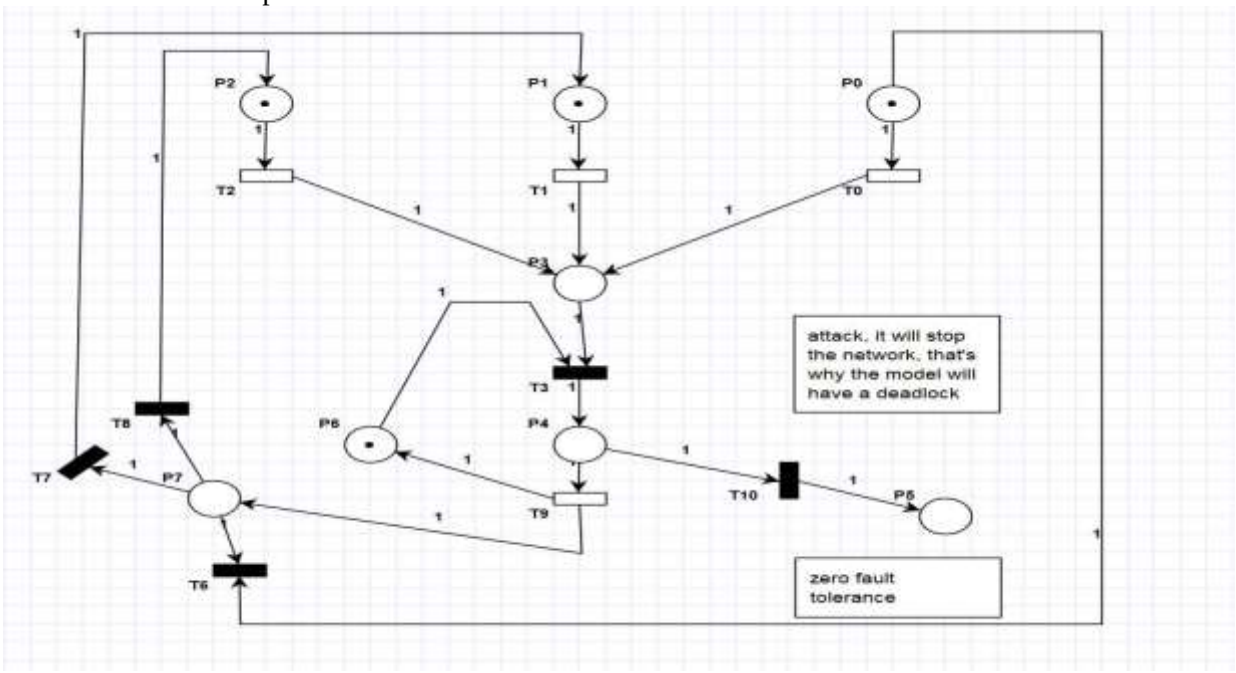

Fig.4. Ordinary topology modelling by using Petri Nets.

Description:

1. This topology is just representing the usual, simple, basic structure of Software-defined network using one controller.
2. It's just modelled for the sake of comparison to show how much effective our framework is with its proposed topologies.
3. This model shows how a one controller is really vulnerable and ineffective since there's a single point of failure (SPOF) which we want to overcome.
4. We have here one controller that processes switches' requests normally until an attack occurs.
5. In the case of an attack the above design shows that an attack can disrupt the controller and everything that relies on it since there's only one main controller. so, everything falls apart after infecting the controller and the whole network will be compromised. Which means that this topology has zero fault tolerance. The table 3 below shows the places of the diagram.

Table 3

Description of Places

Place	Description
P0/P1/P2	Selection of switches
P3	Main controller/server
P4	Active processing
P6	Processing next request/getting back to initial state
P7	Sending and receiving requests
P5	Attack on server/controller

Table 4 describes the transitions of the petri nets diagram.

Table 4

Description of Transitions

Transition	Description
T0/T1/T2	Sending requests to controller
T3	Active processing
T9	Initial state/replying to switches
T6/T7/T8	Selection of switches
T10	Attack

5. Defence Factor Formula: Derived from the Petri Nets modelling

Here we'll try to explain the basis of the defence factor equation formulation and what tools were used for that.

- **GSPN Module**

Before talking about the defence factor formula, there is a need for explaining the system it was based on. A Generalised Stochastic Petri Nets (GSPN) system is a 6-tuple (P, T, F, W, M_0, λ) module where:

- P= {P_1, P_2, …, P_m} is a finite set of places, n ≥ 0.
- T= $T_1 \cap T_2$, T1= {t_1, t_2, …, t_m} is a finite set of timed transitions, and each of these transitions is associated with a random delay time between enabling and firing. And T2= {t_{m+1}, t_{m+2}, …, t_n} is a finite set of immediate transitions, which can be fired randomly, and the delay is zero.
- F⊆ (PxT) ∩ (TxP) is a set of arcs; also, there exit inhibitor arcs that can only also form places to transitions and make the enable conditions be disenabled.
- W is a weight function of arcs: F→{1,2,3, ….}.
- M_0:P → {0,1,2,3, …}is the initial marking, where (PxT)= *Ø* ∩ (TxP)= *Ø*.
- λ= {λ_1, λ_2, λ_3, … λ_n} is a set of firing rates corresponding to the timed transitions. Each rate is the average firing times of transition in unit of time [10].

Here we aim to determine the best topology or the most secure one to create the most reliable and best SDN environment performance with more reliability and ability to deter cyber-attacks like DoS/DDoS attacks.

Next, there is a simple comparison between the two topologies in terms of Average Number of Tokens in the Petri Nets places that represent the SDN controllers in the respective topologies, where the tokens represent the number of tasks or configuration updates the controllers have to execute every 10 seconds and the less busy controllers are, the better it is and the more robust topology it is, because it means that the network controllers are less DoS/DDoS attacks prone and more capable of handling these attacks. In this comparison, we left the weight ω of immediate transitions intact and with no change and gave the rate r of timed transitions a value of 0.1 because those configurations of the network will be broadcasted every 10 seconds which means that every 10 seconds the model state will change. The relationship between the time and rate/weight can be shown as below:

$$\tau = \frac{1}{r} \quad (1)$$

- where τ represents the time, r represents the rate of timed transitions. That's why, if we want the time to be 10 seconds then, we have to change the rate value to 0.1.
- We gave the models a fixed value of firings for the transitions in each model which is 20 firings and the result was as shown in the table 5.

Table 5

Average Number of Tokens in Places Representing SDN Controllers by Using GSPN Module

Algorithms / Places/k_i	Hybrid Topology/ Z_{Ki}	Ordinary Topology/ Z_{Ki}
P3		1.99975≈2
P1	0	
P5	0	
P9	0	
P21	0.9037	
P22	0.90368	
P26	0.90352	

Based on table 5, it is possible to notice that using this topology the average number of tokens which is the average number of processes or requests dealt with by the controller per unit of time is really reduced and small as compared to the average number of token/requests dealt with by the controller in the ordinary topology and that could prove the efficiency of the topology as compared to the efficiency of the single-controller SDN topology.

- **Elaboration of the Defence Factor formula**

According to table 5, it is possible to notice that the Hybrid topology will be better since its model is emptier of tokens most of time in which the firings took place, which means that this network topology was less occupied with tasks or its controllers were more available per unit of time, hence, more capable of resisting DoS/DDoS attacks. Now after using the GSPN module in the PIPE software that provided us with those numerical results mentioned earlier, it is possible to elaborate an equation based on the acquired results and based on the relationship between the readings gained from the different simulations, meaning the equation to assess the security level of networks especially the software-defined networks against cyberattacks, especially those that could use the busy or flooded servers as a weakness point, like DoS/DDoS attacks; we called this proposed equation the network defence factor against cyberattacks law. Before using this law or equation, it is needed to describe the basis of this law itself and how and why it was formulated. First, we have to emphasize that this law is formulated for different kinds of attacks but it is especially used for DoS/DDoS attacks risk assessment due to its main concept or feature that it depends on; which is how many operations are conducted by the controller/server, means how many requests that the controller is dealing with at a specific unit of time and as we know and mentioned before, that denial of service and distributed denial of service attacks DoS/DDoS are depending mainly on flooding the target with huge amounts of request packets to disrupt it or stop it completely so, the less the targeted device is occupied the better it is, because it will be more capable of dealing with that big amount of requests hence, it will have a better security defence level and longer time to respond to use its defence mechanisms like intrusion detection and intrusion prevention systems IDS/IPS, firewalls, etc. The law shows that the less requests a controller has, the better security level and higher defence abilities it has and vice versa; so, it's an opposite relationship between the number of requests being handled at a specific unit of time and the Defence level assessment. And as we know, this research focuses on assuring the security of the computer network generally and the security of the software-defined network particularly, especially the control plane in its structure. And we have the control plane represented by the controller as our main element of interest to secure and also the main component of the SDN that we need to determine its security level, then it is of a great deal of importance to include that element mainly in the formula to figure out its security level, hence, finding out the security level of the network itself.

In other words, let defence factor be DF, then:

$$DF=f\{K, Z\} \tag{2}$$

Where K is the number of controllers in the network, and Z is the number of requests being served in each controller at the current unit of time. If we use the aforementioned relationship with the Petri Net models that we have; then we can get a relationship between all of them which is our aforementioned equation that can be applied using the terms of Petri Nets as well. In terms of petri nets the requests will be represented by how many tokens are there in specific places, which in turn represent specific nodes in the software-defined network and those specific nodes of interest are the SDN controllers. In the equation places representing the SDN controllers are denoted as K, where $K \in P$ which is the whole group of places in the petri nets PN model, which in turn is a tuple of 5 objects, $Pn= (P, T, I, O, M_0)$, where P is the finite set of places, T is a finite set of transitions, I is the input function, then we have O for output function and M_0 is the initial marking. So, here is the equation to measure the defence factor for a software-defined network that is based on one of the 3 specific topologies we proposed previously:

$$DF= \sum_{i=1}^{i=n} k_i \cdot \frac{1}{\sum_{z=0}^{z=\infty} Z_{k_i}} \tag{3}$$

Where K_i as mentioned previously is the number of the places that represent the controllers in a specific model, $K_i= (K_1, K_2, \ldots\ldots K_n)$ and Z_{Ki} is the value of tokens in those places K_i, $Z_{ki} = (0\ldots \infty)$.

$$\text{Let } A = \frac{1}{\sum_{z=0}^{z=\infty} z_{k_i}} \tag{4}$$

$$A = \begin{bmatrix} \infty & ,(Z) = 0 \\ < 1 & ,(Z) > 1 \\ 1 & ,(Z) = 1 \\ 0 & ,(Z) = \infty \end{bmatrix} \tag{5}$$

So, if we apply the values obtained from that simulation table of module on this proposed formula; then, we have the Defence Factor for the ordinary single-controller topology of SDN will be:
$DFO = Ko_i.1/Zo_i = 1x\ (1/T_{P3})$ → where T_{P3} is the value of the average number of tokens in the place No.3 (P3), then
$DFO = 1x1/1.99975 = 0.50$
While the Defence Factor for the proposed Hybrid topology of SDN will be:
$DFH = Kh_i.1/Zh_i = 6x\ (1/\ T_{P1}+T_{P5}+T_{P9}+TP_{21}+T_{P22}+T_{P26})$ → where T_{P1}, T_{P5}, $T_{P9,}$ TP_{21}, T_{P22}, T_{P26} are the values of the average number of tokens in those places respectively as well, then
$DFH = 6x1/2.7109 = 2.21$

It is needed to point out here that the Hybrid topology has a higher defence capability than the Ordinary one. Table 6 below presents a comparison between the different topologies in their Defence Factor results.

Table 6

Comparison between Different SDN Topologies based on Their Defence Factor DF

Algorithm	Hybrid Topology	Ordinary Topology
Defence factor DF	2.21	0.50

Conclusions

- This paper concentrates on assuring the security of software-defined networks in order to make them a safer environment hence, ensuring the security of computer networks in general by facilitating their transition to the SDN paradigm.
- In this paper we've provided a brief explanation of the algorithms proposed by us and that will be explained in later articles. They're incorporated together to form the Hydra framework.
- In this research we have designed this framework to be used optionally by the other part of the solution which is the topologies proposed to overcome the centralisation point in the SDN structure which is an advantage itself since it facilitates the management of the network but, at the same time, it raises some new security challenges as well, like the single point of failure (SPOF).
- The research on which this paper is based on proposes three topologies for the SDN controllers, and this article focuses on one of them which is the Hybrid topology.
- This article provided modelling for the Hybrid and single-controller Ordinary topology using Petri Nets system to derive a formula that can be used to assess the security risk level of any software-defined network that is based on any one of the proposed topologies.
- In short, this paper provides a new approach for assuring the security of software-defined networking using newly proposed SDN controller topologies. Where the novelty of usage of these proposed topologies revolves around the kind and way of the interaction between their controllers.
- This article tries to give a new security assessment equation to assess the threat level and the defence ability of the network that is based on the SDN paradigm and that's done by using Petri Nets modelling approach for those topologies and then based on that modelling, a mathematical relationship from the gained results of the software's simulation was derived.
- A relationship that could fulfil the same implementation of the numerical results was derived here. Based on the gained numerical results it is possible to conclude that the Hybrid topology is better than the ordinary topology and, in the relationship that we have proposed, we gain the same implications which match the gained numerical results.

- The scientific value of the article mainly revolves about the ability to find a measurement tool for the risk level/security level of any network that uses the SDN paradigm and specifically if it was leveraging any of the proposed topologies in this research where we depicted one of them which is the Hybrid topology and compared it with the existing usual one that was named in our research as the ordinary topology.
- The correctness of the proposed formula is derived from its matching to the results gained by the PIPE software simulation, since both the simulation results and the implementation of the formula show that the Hybrid topology is better and its controllers are more free along most the average processing time and that means that they're emptier than the single controller of an ordinary single-controller topology and they'll be more capable of deterring DoS/DDoS attacks that aim to submerge servers with fake requests.

References:

1. AMEEN, A. Software-defined networks - a general survey and analysis. In: *Journal of Engineering Science*, 2018, no.3, p.61-73. ISSN 2587-3474
2. AMEEN, A. The using of sdn technologies for security insurance of computer networks. In: *Proceedings of Technical-Scientific Conference of TUM*, March 2019, Vo.1, p.417-420.
3. AMEEN, A. leveraging blockchain technology to assure security of SDN. In: *Journal of Engineering Science, Proceedings of International conference on Electronics Communications and computing*, 2019, no.4, p.128-139. ISSN 2587- 3474
4. AMEEN, A. Deriving a security formula using petri nets system to assure the security of SDN. In: *Proceedings of Journal of Engineering Science*, to be published.
5. AYESHA, I. *SDN controllers' security issues*: MS thesis. University of Jyväskylä - Finland, 2017.
6. MALIK, A., AHSAN A., SHAHADAT, M., TSOU, J. Man-in-the-middle-attack: Understanding in simple words. International Journal of Data and Network Science, 2019, China University of Technology, Vol.3, p.77-92.
7. 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. Vol.14. 18 p.
8. WONG, F., TAN, C.X. A survey of trends in massive DDoS attacks and cloud-based mitigations. In: *Int. J. Netw. Secur. Appl*. (IJNSA), 2014, 6(3), pp.57-71.
9. ZAKARIA BAWANY, N., A. SHAMSI, J., SALAH, K. DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions, King Fahd University of Petroleum & Minerals, ISSUE 2017. In: *Journal of Cryptology*, 2017. 17 p.
10. ALMUTAIRI, L., SHETTY, S. Generalized Stochastic Petri Net Model Based Security Risk Assessment of Software Defined Networks. In: *IEEE Military Communications Conference*, 2017, p.545-550.
11. OSAGIE, M., ENAGBONMA, O., INYANG, A. The Historical Perspective of Botnet Tools. In: *Current Journal of Applied Science and Technology*, Issue 6, Feb, 2019, Department of Physical Sciences, Faculty of Science, Benson Idahosa University, vol.32. 8 p.
12. CCTV-based botnet used for DDoS attacks. [Online]. [Accessed: 04.07.2017]. Available: https://www.ddosattacks.net/a-massive-botnet-of-cctv-cameras-involved-inferocious-ddos-attacks
13. IQBAL, M., RIADI, I. Analysis of Security Virtual Private Network (VPN) Using OpenVPN. In: *International Journal of Cyber-Security and Digital Forensics*, 2019, Ahmad Dahlan University, no.8(1), p.58-65.
14. RSA. [Online]. 2019, [Accessed: 20.07.2019] available: https://en.wikipedia.org/wiki/RSA_(cryptosystem)
15. How blockchain will manage networks. [Online]. 2019, [Accessed: 26.08.2019] Available: https://www.networkworld.com/article/3356496/how-blockchain-will-manage-networks.html
16. Petri Net. [Online]. 2019, [Accessed: 11.10.2019] Available: https://en.wikipedia.org/wiki/Petri_net

Data about author:
Ali AMEEN, Ph.D. student, Technical University of Moldova.
E-mail: alisalmanhussein@yahoo.com
ORCID: 0000-0002-5451-8257

Prezentat la 29.04.2021