Security-Oriented Network Intent Placement using Particle Swarm Optimization

Gabriel Landeau*, Marios Avgeris†, Aris Leivadeas*, Ioannis Lambadaris†
*Department of Software and IT Engineering, École de technologie supérieure (ÉTS), Montréal, Canada
†Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada
gabriel.landeau.1@ens.etsmtl.ca, mariosavgeris@cunet.carleton.ca, aris.leivadeas@etsmtl.ca, ioannis@sce.carleton.ca

***Abstract*—As the network infrastructure grows, its configuration and service provisioning become a tedious process. Accordingly, new paradigms have emerged, such as the Intent-Based Networking (IBN), that envision the automation of the network configuration, while minimizing the human intervention. Specifically, IBN allows users to interact with the network through high-level and declarative requests, called intents, which later can be translated into low-level configurations. IBN can entail different scopes and target network infrastructures, while being domain specific, which can create several challenges in terms of the final activation of the requested intents. To this end, in this paper, we mainly focus on intents that are expressing security and Quality of Service (QoS) network services demands that can be translated into Service Function Chains (SFC) and automatically deployed over a campus network. Our work and depending on the security level expressed in the intent, tries to optimally decide the level of multi-tenancy or complete segregation of the users' services that can be achieved, while satisfying the network provider's objectives. In particular, an artificial intelligence inspired algorithm called Particle Swarm Optimization (PSO) is modeled that automatically tries to find the best placement of the intents, while satisfying the security and QoS requirements of the users issuing the intents.**

***Index Terms*—Intent-based Networking, Network Function Virtualization, Particle Swarm Optimization, Cybersecurity**

I. Introduction

Intent-Based Networking (IBN) is a novel network paradigm that envisions replacing error-prone manual network configurations, with high-level and abstract network requirements, called intents [1]. These intents, through a closed loop automation process, will be translated into low-level configurations and finally activated and assured on top of the network fabric [2]. IBN is expected to be based on recent advances on network softwarization, such as Network Function Virtualization (NFV), Service Function Chaining (SFC), and Software Defined Networking (SDN). For instance, the European Telecommunications Standards Institute (ETSI) standardizes Network Function Virtualization (NFV), encompassing virtualized network functions (VNFs) that dynamically implement user intents [3].

IBN can alleviate the hectic manual configuration of network configurations which resulted from the unprecedented growth of the network infrastructure and its users. However, the complexity of network configurations is not the sole repercussion of the surging user count; the cybersecurity attack surface gets wider as well [4]. To mitigate this phenomenon, the NFV framework is again offered as a viable solution by enabling the specialization of network functions, particularly in the domain of security. By augmenting the knowledge base of network functions with security capabilities, this improvement yields two significant impacts: tailored security services aligned with user intents and the feasibility of executing these services on bare-bone server infrastructures [5].

Furthermore, the NFV framework empowers fine-grained control over server configurations and logical micro-segmentation, fostering enhanced isolation of customers in both the physical and logical layers. This attribute effectively enhances privacy protection while facilitating multi-tenancy across the network and within the server infrastructures. Nevertheless, the pursuit of physical isolation may necessitate an increase in the number of servers required to fulfill similar intents, albeit offering a secure multi-tenancy environment where shared resources can serve different clients. This can also come in contrast with the infrastructure provider's requirements on how the servers will be used, since a large number of active servers may introduce significant operational and energy consumption costs.

The lack of related works on implementing IBN for resolving network and security placement issues in a multi-tenancy NFV framework serves as the motivation for this research. Therefore, this paper aims to address the optimization of network and security function placement in a campus network using IBN and NFV, offering a human-centered and flexible solution to meet user and infrastructure provider's requirements. From the user point of view, the proposed solution aims to fulfill users' security, network, and QoS needs, while for the provider to optimize the resource utilization and to reduce physical dependencies through virtualization. However, this optimization problem poses significant challenges due to numerous constraints and the vast number of feasible solutions which constitute it a computationally intractable NP-Hard problem. Thus, stochastic optimization-based algorithms, known for their real-time problem-solving capabilities, are deemed more suitable. In this regard, the proposed solution employs a customized version of the Particle Swarm Optimization (PSO) algorithm, a neighbor-based learning algorithm, to effectively address this complex optimization problem in real time. The contribution of this work is threefold:

1) We study the intent expression for security services in a campus network that can be translated into security VNFs, while taking into consideration their QoS require-

ments.

2) Then, we propose a multi-tenancy aware VNF placement solution optimized to minimize resource consumption in terms of activated servers in the network.
3) Through extensive simulation we demonstrate the efficiency and efficacy of the proposed solution, when compared to other baselines.

The rest of the paper is organized as follows: Section II discusses the related works in the literature. Section III introduces the modeling of the problem and proposes a solution based on the PSO algorithm. The experimental results are presented in Section IV. Finally, Section V concludes the paper.

II. Related Work

The existing research in the field lacks comprehensive investigations into the intent activation and VNF placement optimization while considering security and QoS requirements in a shared resource environment where SFCs operate. In [6], the authors present an initial approach called INSpIRE, which proposes a method to activate intents using service chaining with both VNFs and dedicated physical servers. The solution decomposes intents into smaller security attributes and implements them as virtual functions and specialized devices, such as firewalls. In [7], a virtual network management platform is introduced to facilitate network administration through intents. This solution, based on the SDN ONOS controller, enables the provisioning of virtual networks that share resources to meet multi-tenancy constraints. Addressing the multi-tenancy challenge, [8] proposes a solution that translates intents by mapping each intent to a VNF blueprint in a knowledge base. The most appropriate chaining service is determined based on these blueprints. To address intents in cloud computing scenarios, [9] proposes a solution that combines an IBN platform for intent translation with an NFV framework for intent activation. This solution enables the provisioning of user security and network requirements while emphasizing on the optimization of VNF placement on a cloud infrastructure, rather than considering the security services provided. In an effort to enhance security by providing multi-tenant services, [10] introduces an optimized placement solution for resource consumption in cloud computing. This solution achieves isolation by allocating dedicated servers to customers, thereby preventing unauthorized access by users outside the designated scope. However, the above approaches lack research on logical isolation to offer fine-grained security and do not fully explore the potential of IBN for improving user experience. To deal with these shortcomings, this article presents a placement optimization solution leveraging IBN and software-defined techniques to offer multi-tenant services. The proposed solution focuses on addressing security and QoS considerations while minimizing resource consumption within a network campus environment. By harnessing virtual network and security functions, our approach facilitates the provisioning of diverse services, including logical and physical isolation, to cater to the intents of campus network users.

III. System Model and Algorithm Design

A. IBN Model

The user can interact with the IBN system of the campus by expressing a type of application, as well as a security and QoS level. In this paper, we do not concentrate on the way that the user expresses her intent (i.e., through a GUI, natural language, etc.), but rather on the high-level content of the intent. For this reason, we assume that the user can request a high-level intent specifying a level of security that can range from 1 to 4, with 1 being the weakest and 4 being the strongest level. Similarly, for the QoS level, the user can express a qualitative value such as *irrelevant*, *best-effort*, and *relevant*.

Each level of security will be associated with a pre-configured SFC in terms of which VNFs to be used and with what order. Herein, the available VNFs are virtual routers, Firewalls (FW), Intrusion Detection Systems (IDS), Deep Packet Inspection (DPI), and three different types of encryption (i.e. PHYSec, MACSec, IPSec). The weaker the level of the security intent the lower will be the number of security VNFs in the SFC (i.e., only virtual router and a firewall for the first level of security). Additionally, for the strongest level of security (i.e., 4), there will not only be more security VNFs on the SFC, but also the IBN System will add the requirement of complete isolation for that particular intent. This means that for the intent asking for a level 4 security, the VNFs allocated to it will be only used by that particular intent and not shared among different intents. This necessitates instantiating each VNF of the respective SFC on servers exclusively dedicated to that specific SFC, or potentially on servers exclusively utilized by the user expressing the intent.

Providing services with specific resource requirements and isolation incurs higher operational costs for the provider due to the obligation of reserving physical servers and links. Thus, to achieve a balance between user intent satisfaction and infrastructure costs, we aim to optimize the placement of SFCs of security level 1 to 3 to reap the benefits of multi-tenancy and ensure the continuous server availability for fulfilling level 4 requests in a resource-constrained infrastructure. The QoS is also a constraint of this optimization problem, which ensures that the activated intent complies with the user requirements.

B. PSNIP: PSO-based Secure Network Intent Placement

In general, the VNF/SFC placement problem is known in the literature to be an NP-hard problem [11]. This means that the execution time of a mixed-integer programming solver increases exponentially with the size of the infrastructure, hence the importance of heuristic algorithms which can provide near-optimal solutions in much faster time, without being impacted by the infrastructure size, is evident.

Thus, we present our heuristic solution, *PSNIP:* a *PSO-based Secure Network Intent Placement* algorithm (Algorithm 1). In our case we define as a particle a potential feasible placement for the given intents, which consists of an embedding of a virtual graph composed of the activated, interconnected VNFs to the physical graph (physical servers connected with

physical links) representing the infrastructure. The position vector of the *i-th* particle is expressed as x_i and, naturally, the larger the particle population size, the better the accuracy of the algorithm in the expense of a slower convergence and vice-versa. For each particle a velocity matrix is also defined as $v_i = (v_{i,1}, v_{i,2}, ..., v_{i,s}, ..., v_{i,S})$, $s \in S$ denoting the server, giving the algorithm a direction for the changes to implement through the iterations, i.e., if $v_{i,s} < 0$ then $v_{i,s}$ VNFs in number need to migrate from server s, otherwise $v_{i,s}$ VNFs can be migrated to that server. In reality, we implement , so the following conditions have to be met: i) the VNF type has to be present at the target server, ii) physical connections that connect the target server to the servers that contain the previous and the next VNFs of the examined SFC have to be present and iii) the security level/isolation requirements have to be met. After randomly initializing the particles, the velocity, and the position of the *i-th* particle on the *(t+1)-th* generation are updated as:

$$v_i^{t+1} = w v_i^{t+1} + c_1 r_1 (p_i^t - x_i^t) + c_2 r_2 (p_g^t - x_i^t), \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \quad (2)$$

where, $w \in [0.5, 1]$ is the inertia weight used to control the influence of previous velocity on the new one; parameters c_1 and c_2 determine the weights of p_i and p_g, which represent the best previous position of the *i-th* particle and the best previous position of all particles in the current generation respectively; r_1 and r_2 are random values uniformly distributed in $[0, 1]$. The

Algorithm 1: PSNIP

```
Initialize population either randomly or greedily.
for t = 1 : maximum_generation do
    for i = 1 : population_size do
        if f(x_i^t) < f(p_i^t) then
            p_i^t ← x_i^t
            f(p_g^t) ← min_t f(p_i^t)
        end
    end
    for s = 1 : S do
        Update v_{i,s}^{t+1} and x_{i,s}^{t+1} using (1) and (2).
        Migrate VNFs while respecting constraints.
        if v_{i,s}^{t+1} > v_max then v_{i,s}^{t+1} = v_max
        else if v_{i,s}^{t+1} < v_min then v_{i,s}^{t+1} = v_min
        if f(x_{i,s}^{t+1}) > f(x_max) then x_{i,s}^{t+1} = x_max
        else if f(x_{i,s}^{t+1}) < f(x_min) then x_{i,s}^{t+1} = x_min
    end
end
```

evaluation function $f(\cdot)$ quantifies i) the number of servers utilized and ii) the intent drop ratio of the particle. For the initialization of the particles, we experimented with two alternatives: i) a random feasible initial placement (RPSO) and ii) a greedy one where the already activated servers and their closest neighbors are examined first for the placement of an SFC (GPSO). The maximum generation limit is selected empirically.

IV. Performance Evaluation

For the evaluation, we perform a series of comparative experiments between the proposed mechanism's both initial placement implementations and the baseline cases of random and greedy placement without the PSO optimization. We consider an infrastructure consisting of $S = 200$ servers and 120 user intent requests. The population size is equal to 60, the maximum number of generations equal to 100 and $c_1 = c_2 = c = 0.8$, for a more balanced exploration vs. exploitation outcome, unless stated otherwise. The results are averaged over 100 executions for each experiment family.

Regarding the impact of the particle population size, we observe in Fig. 1a, that the more the particles, the more efficient the minimization of activated servers. A similar behavior is observed for the number of maximum generations (Fig. 1b). This comes naturally as more possible solutions are examined and a greater number of generations is allowed respectively, in the expense, however, of an increased complexity and execution time. In both cases, the greedy initialization outperforms the random one. As for the velocity coefficients $c_1 = c_2 = c$ (Fig. 1c), the higher their value is, the slower, albeit more accurate, the convergence to a stable solution is, and vice-versa.

Figs. 1d-1g showcase the benchmarking of our algorithm against the baseline solutions. Specifically, in Fig. 1d we observe that when the intent placement problem is no longer trivial, i.e., the solution search space becomes large, that is when the proposed solution yields meaningful results. For our setting, when the infrastructure grows larger than 75 physical servers, the PSO-based solutions start to outperform the baseline ones, with the GPSO once again achieving the minimum number of activated servers. We should note here that until this point, all four solutions report a significant drop rate, ranging from $\sim 40\%$ to $\sim 5\%$ for the PSO-based ones and $\sim 65\%$ to $\sim 10\%$ for the baseline ones, as we move closer to the sweet spot of 75 physical servers where the drop rate becomes zero. This is a result of the inability to place the intents while respecting the security/isolation constraints in a small infrastructure.

Next, we evaluate the impact of the infrastructure size on the server utilization, which in this case is calculated as a weighted sum of CPU and memory utilization. In detail, in Fig. 1e we see that the proposed solution tends to collocate more VNFs in the same physical servers, to keep the objective cost of activated servers low, while the non-PSO-based solutions spread them. Once again, the greedy particle initialization results in a more efficient placement with higher utilization on the activated servers. As previously, the benefits kick in when the infrastructure size is larger than 75 physical servers. We should also note here that for this family of experiments the average execution time for the PSO-based solutions went from $< 1s$ when the infrastructure consisted of less than 100 physical servers, to $> 10s$ when this number doubled, without significant improvements in the cost minimization.

On the other hand, Fig. 1f shows how increasing the number of intent requests affects the efficiency of the four solutions.

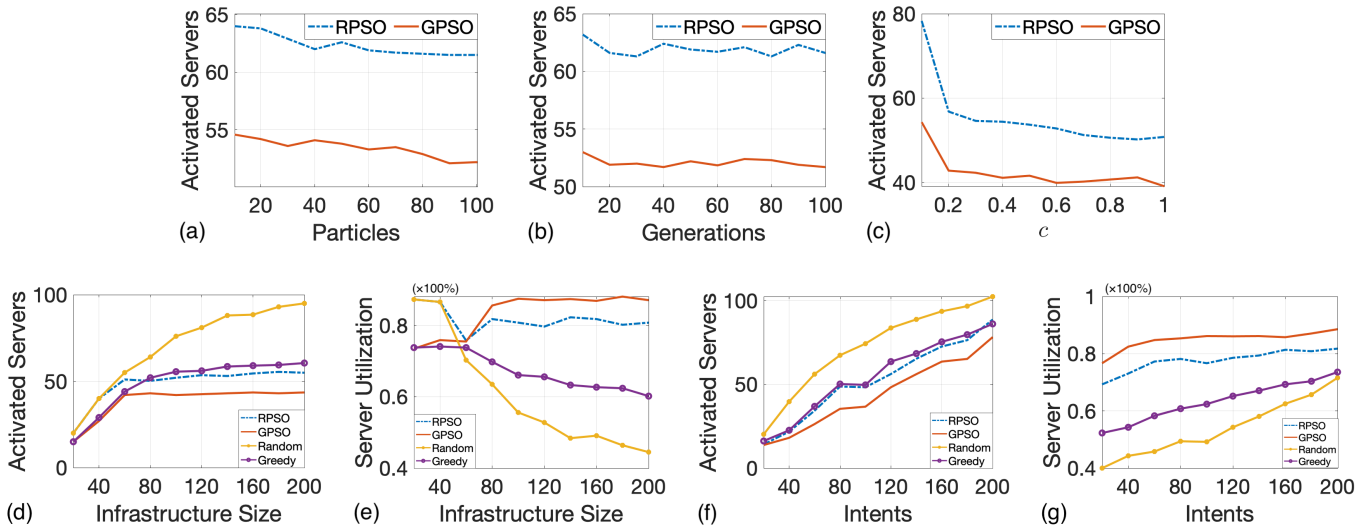

Fig. 1: Performance Evaluation

As expected, the number of activated servers is increased too to adjust to the demand, and it does so in a linear-like fashion. A similar trend is observed in Fig. 1g where the server utilization is displayed. In both cases, the PSO-based solution, specifically the one using the greedy particle initialization, performs best, always resulting in the most cost-efficient intent placement that respects the security constraints.

V. Conclusion

In this paper we dealt with the problem of user network intent instantiating in the form of SFC placement. Specifically, we investigated the problem of incorporating the various security and isolation requirements of the user intents on a university campus scenario, while keeping the computational complexity of the solution low. To this extend, we proposed a PSO-based Secure Network Intent Placement algorithm which takes advantage of the native neighbourhood-oriented optimization to iteratively minimize the number of activated servers that are used to host the SFCs required to implement the intents. The preliminary obtained results indicated that the proposed approach outperforms the baseline ones, by providing a more efficient and low-cost solution. As future work, we are currently studying and working on integrating additional particle initialization algorithms, as we showed that this step affects the outcome in a non-trivial degree. Additionally, we plan to actively include the execution time of our proposed solution as an optimization criterion to provide a more realistic approach for a real-world implementation.

References

[1] A. Clemm, L. Ciavaglia, L. Z. Granville, and J. Tantsura, "Intent-Based Networking - Concepts and Definitions," RFC 9315, Tech. Rep. 9315, Oct. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9315

[2] A. Leivadeas and M. Falkner, "A survey on intent-based networking," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 625–655, 2023.

[3] T. Szyrkowiec, M. Santuari, M. Chamania, D. Siracusa, A. Autenrieth, V. Lopez, J. Cho, and W. Kellerer, "Automatic intent-based secure service creation through a multilayer sdn network orchestration," *J. Opt. Commun. Netw.*, vol. 10, no. 4, pp. 289–297, Apr 2018.

[4] B. Tian, X. Zhang, E. Zhai, H. H. Liu, Q. Ye, C. Wang, X. Wu, Z. Ji, Y. Sang, M. Zhang, D. Yu, C. Tian, H. Zheng, and B. Y. Zhao, "Safely and automatically updating in-network acl configurations with intent language," in *Proceedings of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 214–226. [Online]. Available: https://doi.org/10.1145/3341302.3342088

[5] S.-Y. Wang and Y.-H. Hsieh, "Running an ids virtual network function inside an sdn bare metal commodity switch," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[6] E. John Scheid, C. Machado, M. Franco, R. Santos, R. Pfitscher, A. Schaeffer-Filho, and L. Granville, "Inspire: Integrated nfv-based intent refinement environment," 05 2017, pp. 186–194.

[7] Y. Han, J. Li, D. B. Hoang, J.-H. Yoo, and J. W.-K. Hong, "An intent-based network virtualization platform for sdn," *2016 12th International Conference on Network and Service Management (CNSM)*, pp. 353–358, 2016.

[8] G. Davoli, W. Cerroni, S. Tomovic, C. Buratti, C. Contoli, and F. Callegati, "Intent-based service management for heterogeneous software-defined infrastructure domains," *International Journal of Network Management*, vol. 29, no. 1, p. e2051, 2019, e2051 nem.2051. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.2051

[9] A. Leivadeas and M. Falkner, "Vnf placement problem: A multi-tenant intent-based networking approach," in *2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2021, pp. 143–150.

[10] J. Gao, L. Feng, P. Yu, F. Zhou, Z. Wu, X. Qiu, J. Li, and Y. Zhu, "Resource consumption and security-aware multi-tenant service function chain deployment based on hypergraph matching," *Computer Networks*, vol. 216, p. 109298, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622003516

[11] S. Deng, Z. Xiang, J. Taheri, M. A. Khoshkholghi, J. Yin, A. Y. Zomaya, and S. Dustdar, "Optimal application deployment in resource constrained distributed edges," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1907–1923, 2020.