NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium | 978-1-6654-7716-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/NOMS56928.2023.10154316

Model Predictive Control for Automated Network Assurance in Intent-Based Networking enabled Service Function Chains

Marios Avgeris *, Aris Leivadeas †, Nikolaos Athanasopoulos ‡, Ioannis Lambadaris *, Matthias Falkner §
* Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada
Email: mariosavgeris@cunet.carleton.ca, ioannis@sce.carleton.ca
† Department of Software and IT Engineering, École de technologie supérieure, Montreal, Canada
Email: aris.leivadeas@etsmtl.ca
‡ School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast, UK
Email: n.athanasopoulos@qub.ac.uk
§ Cisco Systems, Ottawa, Canada, Email: mfalkner@cisco.com

***Abstract*—Recent trends in Network Function Virtualization (NFV) combined with Internet of Things (IoT) and 5G applications have reshaped the network service offering. In particular, Service Function Chains (SFCs) can associate network functions with physical and virtual resources towards providing a complete network service. Concurrently, the management of a continuously expanding network and the fulfillment of the applications' requirements pave the way for autonomic network solutions. Intent Based Networking (IBN) is a novel paradigm that aims to achieve the automatic orchestration of network services and the assurance of their performance. Accordingly, in this paper, we propose a novel automated network assurance model, based on Model Predictive Control, to guarantee the Quality of Service (QoS) and security requirements of multi-tenant and IBN-enabled SFCs. In this context, corrective decisions are proactively taken, in the form of incoming intent relocations among the SFCs. The results reveal that our model can assure with high probability the application requirements and minimize QoS violations.**

***Index Terms*—Intent Based Networking, Service Function Chaining, NFV, Model Predictive Control**

I. Introduction

As the network infrastructure grows and new applications emerge, it becomes strenuous to manage the network. At the same time, new applications with strict Quality of Service (QoS) requirements necessitate a stable communication with performance guarantees. Therefore, the network should be able to adjust to any performance drift that may result to any requirements violations. In this Network Assurance problem [1], the performance degradation must be promptly detected and appropriate network actuators should be triggered.

Obviously, reforming the network is not an effortless task. On top of that, error prone manual configurations are no longer a viable option for large networks. Thus, new autonomic management approaches need to be deployed to automate this network assurance. Intent Based Networking (IBN) is such an approach that envisions creating a self-sustained network. IBN gives the opportunity to the users to express their high-level requirements in a declarative way, called intent [2]. A simple example can be "User A wants to talk to User B through a video conference application with high QoS and default security". Following, this intent can be translated into specific network configurations and installed in the network fabric. Finally, the deployed intent is monitored in order to detect any deviations from the initial requirements.

This process creates a Closed-Loop Automation (CLA) system that relies on the intent translation, activation and assurance processes [3]. Often, the intent is translated into a Service Function Chain (SFC) that is composed of a number of Virtualized Network Functions (VNFs) [4]. The translated intent can be later activated by using an appropriate VNF placement solution that allocates the SFC to the underlying physical infrastructure [5]. The allocation can take into consideration the requirements of the user by appropriately constructing its objective function. For instance, when the intent requests a high QoS, the objective function could be formulated as to maximize the throughput or minimize the delay of the communication.

However, this will only guarantee that the QoS will be satisfied upon the activation of the intent. In contrast, the network or the intent itself can dynamically change throughout its lifetime leading to performance uncertainty. To this purpose, in this paper, we propose a new network assurance model in the form of a discrete time dynamical system, where the QoS, specifically end-to-end delays, are captured explicitly as time varying states. Subsequently, using tools from Model Predictive Control (MPC) [6] and optimization, we are able to formulate the problem of satisfying the user intents (user intents and users will be used interchangeably) at all times, as a finite optimization problem which we solve in real time. In more detail, we translate the intent into an SFC with specific security and QoS requirements that may be requested by multiple users at the same time, following a multi-tenant model. The number of users associated with an SFC can affect

Work supported in part by the Cisco University Research Program Fund of Silicon Valley Community Foundation 2021-234759 (3696), and CHIST-ERA CHIST-ERA-18-SDCDN-003 / EPSRC EP/T021942/1 (DRUID-NET).

its overall performance, triggering automatically corrective actions. These actions lead to different assignments of incoming user flows between different SFC instances, in order to satisfy a lower performance bound, while minimizing any QoS and security violations.

The remainder of the paper is structured as follows. Section II highlights the related work. Section III presents the system model and the main components of the network assurance controller. The results and efficiency of our framework are demonstrated in Section IV. Finally, Section V concludes our work and provides some possible future directions.

II. Related Work

In IBN, network assurance can be performed through either reactive or proactive actions. Regarding reactive assurance the most common approaches include flow migration, Virtual Machine (VM) migration, or resource scaling. For instance, the authors in [7], through a Software Defined Networking (SDN) controller periodically monitor the delay in a network path. When a low quality is noticed, the controller selects a different path that can satisfy the intent's QoS. Another approach [8], concerns the use of a k-shortest path algorithm. In particular, a path is selected among k alternatives and if any security (i.e., blocked hosts or links) or QoS conflicts arise with respect to the intent, then a different path is selected.

In the above works, the emphasis is placed on a simple point-to-point connectivity scenario. However, an intent may refer to an SFC. In this case, the monitoring can be shifted at the computational resource consumption of the VM. For example, the authors in [9] periodically poll the status of a VNF (e.g., CPU utilization) and when an overprovisioning event is noticed, the VM migration process is triggered. Alternatively, the whole SFC can be migrated, if the service does not respect the high-level expressed QoS, as in [4]. Another possible choice is to simply scale up the allocated resources to the VMs that host the VNFs of an SFC as proposed in [10].

With reference to proactive IBN assurance, the pertinent literature is mostly divided into flow and VM migration. For instance, the authors in [11] propose a Reinforcement Learning (RL) technique that learns the capacity of a link based on current and predicted traffic sizes. If the traffic and the capacity show opposite trends, then corrective actions can be initiated. Alternatively, taking advantage of the quasi-periodicity of network traffic, the authors in [12] propose the use of historical data to train a routing algorithm to make re-routing decisions for the following time periods. Historical flow data can also be used to predict future Service-Level Agreement (SLA) violations that will result into a set of corrective actions, such as re-routing and flow migration, as in [13].

In the case of resource scaling, the authors in [14] propose the use of a Multilayer Perceptron Neural Network to predict the CPU utilization of a VM hosting a VNF and to trigger a scale up if the predicted utilisation exceeds the acceptable thresholds. Other Neural Networks such as Long Short-Term Memory (LSTM) models can also be used to predict the resource utilization of the VMs for the resource scaling purposes such as in [15]. Except scaling the VM resources, it could also be beneficial to scale the bandwidth resources of virtual links. Accordingly, the authors in [16] leverage historical logs of packet loss and delay and propose the use of an RL technique to scale up the allocated bandwidth.

In this work, our goal is to deal with the challenges posed by the dynamic nature of the IBN driven QoS satisfaction, by proposing, for the first time, an MPC-based framework for proactive, automated network assurance. Specifically, we first utilize an identification method based on constrained least squares, to obtain a linear approximation of the dynamic operation of the SFCs and their interconnections (i.e., relocations of incoming user flows). Having such explicit models that associate user flows with QoS, we are able to formulate the problem of guaranteeing SLA satisfaction as a decision problem. The adopted MPC formalism can handle all the types of constraints that are implicitly or explicitly imposed from the system and its requirements. These include the constraints posed by the QoS and security intents, the finite resources of the SFCs, the discrete nature of the scheduling/relocation problem, and the time-varying uncertainty in the arrival of the incoming user flows; in our formulation, the latter leads to the control inputs (i.e., the variables deciding the flow relocations) depending on time-varying exogenous signals.

III. Control-based Proactive IBN assurance

A. System Model

We consider S levels of security and D levels of QoS intents respectively, which lead to $S \times D = N$ different SFC placements. Each SFC is unique in terms of QoS and security level combination. An index variable $i \in [1, ..., N]$ is used to label the SFCs and the level of offered QoS and security of each one of them is calculated by $d_i = (i-1) \ \mathbf{mod} \ D + 1$ and $s_i = (i-1) \ \mathbf{div} \ S + 1$ respectively; $\mathbf{mod}$ is the modulo operator and $\mathbf{div}$ is the integer division. Towards achieving proactive IBN-assurance, we follow the work in [17] where time is considered slotted. At the beginning of each slot t, we accommodate the partial relocation of incoming user flows among the SFCs. It should be noted that relocation refers to a different assignment of an incoming user from their desired SFC to a different one, since the allocation of the particular user to the requested SFC would negatively affect the existing users that are being already served. Accordingly, for the rest of the paper we will use the terms different assignment and relocation interchangeably. To avoid SLA violations, user relocations are permitted only from an SFC with a lower level of security to an SFC with higher ones. We model the operation of each SFC i as a discrete Linear Time-Invariant (LTI) system with exogenous inputs of the form

$$x_i(t+1) = a_i x_i(t) + b_i u(t) + c_i(v_i(t) - q_i(t)). \quad (1)$$

In Eq. (1), $x_i(t), i \in [1, ..., N]$ is the experienced end-to-end delay, which reflects the achieved QoS and $u(t) = [u_{1,2}(t), u_{1,3}(t), ..., u_{1,N}(t), u_{2,3}(t), ..., u_{N-1,N}(t)]^\intercal \in \mathbb{Z}^{L \times 1}$ is the input vector consisting of the permitted relocations of the incoming, newly associated users among the SFCs, at time

t. Specifically $u_{i,j}(t) \in u(t)$ is a non-negative integer that signifies the number of relocated incoming users from SFC i to SFC j, where $i < j,\ i, j \in [1, ..., N]$. The latter condition defines the permitted relocations, as well as the size L of $u(t)$ which can be calculated using the arithmetic progression as $L = \frac{N(N-1)}{2}$. The exogenous signals $v_i(t) \in [0, V^{max}]$ and $q_i(t) \in [0, Q_i^{max}(t)]$ are unknown, however bounded signals accounting for the total number of incoming and outgoing users respectively. V^{max} denotes the maximum number of users that can be accommodated at the same time in the infrastructure. Also, $Q_i^{max}(t) = \sum_{t'=0}^{t-1} v_i(t')$ is the number of users that stop using an SFC at time t and it cannot exceed the sum of the users that were associated with it until $t-1$.

The parameters $a_i, c_i \geq 0$ are scalars, while $b_i = [b_{1,2}^i, b_{1,3}^i, ..., b_{1,N}^i, b_{2,3}^i, ..., b_{N-1,N}^i] \in \mathbb{R}^{1\times L}$ is a row vector, $i \in [1, ..., N]$. To estimate these parameters, we employ the Least Squares with Linear Constraints and Bounds (LSLCB) algorithm [18], once and offline, on data acquired by the SFCs' operation. This algorithm is preferred because it allows to define additional requirements and constraints that derive from the physical meanings of the unknowns, i.e., a_i, b_i, c_i; in our case, the end-to-end delay experienced in an SFC should remain unchanged if no relocations occur and there is no incoming and outgoing users, i.e., $a_i = 1$. Additionally, relocations of users from SFC i to j, should decrease the delay experienced in i and increase the delay in j. Relocations that happen among SFCs other than SFC i, do not have an impact on its delay. Hence, $b_{i,j}^i \leq 0$, $b_{i,j}^j \geq 0$, and $b_{i',j}^{i'} = 0$ otherwise. Finally, the incoming user flow increases the delay and the outgoing workflow decreases it, meaning that $c_i \geq 0$. Summarizing, after collecting time-series measurements of length T, for x_i, u, v_i and q_i, by relocating user flows and observing the SFC i's operation, we define the following auxiliary constants (Ψ, Ω) and variable (Θ):

$$\Psi = [x_i(1)\ \ ...\ \ x_i(T)]^\intercal,$$

$$\Omega = \begin{bmatrix} x_i(0) & ... & x_i(T-1) \\ u_{1,2}(0) & ... & u_{1,2}(T-1) \\ \vdots & & \\ u_{N-1,N}(0) & ... & u_{N-1,N}(T-1) \\ v_i(0) - q_i(0) & ... & v_i(T-1) - q_i(T-1) \end{bmatrix}^\intercal,$$

$$\Theta = [a_i\ \ b_{1,2}^i\ \ ...\ \ b_{N-1,N}^i\ \ c_i]^\intercal.$$

Then, the parameter a_i, b_i and c_i estimation problem for SFC i can be defined as follows:

$$\min_{\Theta} \quad ||\Psi - \Omega \cdot \Theta||_2^2 \tag{2a}$$

$$\text{s.t.} \quad a_i = 1, \tag{2b}$$

$$b_{i,j}^i \leq 0,\ \forall b_{i,j}^i \in b_i,\ i \leq j, \tag{2c}$$

$$b_{j,i}^i \geq 0,\ \forall b_{j,i}^i \in b_i,\ i \geq j, \tag{2d}$$

$$b_{i',j}^i = 0,\ \forall b_{i',j}^i \in b_i,\ i \neq i',\ i' \leq j, \tag{2e}$$

$$c_i \geq 0. \tag{2f}$$

Furthermore, (1) constitutes a positive system, since x_i corresponds to positive delay times. For example, we examine the simple case where no relocations take place. Then, at time t, $\sum_{t'=0}^{t} q_i(t') \leq \sum_{t'=0}^{t-1} v_i(t')$ and we get:

$$\begin{aligned} x_i(t+1) &= x_i(t) + c_i v_i(t) - c_i q_i(t) = \\ &= x_i(t-1) + c_i v_i(t-1) - c_i q_i(t-1) \\ &\quad + c_i v_i(t) - c_i q_i(t) = \\ &= x_i(0) + c_i v_i(t) + c_i(\sum\nolimits_{t'=0}^{t-1} v_i(t') - \sum\nolimits_{t'=0}^{t} q_i(t')), \end{aligned}$$

which for $x_i(0) = 0$ and based on constraint (2f) gives that $x_i(t+1) \geq 0$. The state variable x is also inherently constrained due to the SFC's capabilities, while the sum of the input variables $u_{i,j}$ is physically constrained by the incoming flow of users v_i of SFC i. Specifically, for all $t \geq 0$,

$$\mathbb{X} := \{x_i : 0 \leq x_i(t) \leq X_i^{max}, i \in [1, ..., N]\}, \tag{3}$$

$$\mathbb{U} := \{u : 0 \leq \sum\nolimits_{j=1}^{N} u_{i,j}(t) \leq v_i(t), i \in [1, ..., N]\}. \tag{4}$$

Based on the above, one can consider the N scalar systems of the form (1) as a large interconnected system, where the coupling is present via the common input constraints, in turn depending on the signals $v_i(t)$ (4). This formulation paves the way for the MPC algorithm to jointly perform the tasks of user relocations and admission control for each SFC, while respecting the SLA constraints, under varying incoming and outgoing user flows. Relocating user flows introduces operational overhead to the infrastructure, so we aim at minimizing the number of relocations at the same time. Our goal is summarized as follows: Consider a system with N state variables of the form (1), subject to constraints (3), (4). Given a desired maximum end-to-end delay for each SFC, $\mathcal{X}_i^{SLA} \in \mathbb{X}$, compute an admissible control strategy, $u \in \mathbb{U}$, so that the constraints are satisfied at all times and the additional labour is minimized.

B. Placement Algorithms

In this work, we follow the placement methodology of [4]. In particular, a hybrid placement solution is followed that accounts for the different and high-level QoS requirements of an intent, which can be "High" ($d = 3$), "Medium" ($d = 2$) or "Low" ($d = 1$). For instance, when a user indicates that a "High QoS" is needed, an optimal placement algorithm is executed [19]. For "Medium QoS" a sub-optimal but efficient placement is performed through an iterative refinement local search (IRLS) [4]. Finally, for "Low QoS", a random and under best effort placement algorithm is executed that places the VNFs of an SFC in a random server that can however guarantee the computational requirements and the communication demands of the SFC. This hybrid approach can configure the network according to the intent and actually deliver an SFC that complies with what was requested in the intent. Regarding the security, three levels are considered, similar to QoS, namely "High" ($s = 3$), "Medium" ($s = 2$), and "Low" ($s = 1$). These levels will have an impact on the structure and length of the SFC. For example, for a "Low Security" a two-VNF SFC is considered that offers basic security features (i.e., a virtual router followed by a firewall). When the user

requests a "Medium Security" a Deep Packet Inspection (DPI) function is added to the SFC. Finally, for a "High Security" the SFC length becomes 5 by adding an Intrusion Detection System (IDS) and an IP encryption function (IPSec).

Furthermore, our approach follows a multi-tenant strategy to maximize the benefits of NFV and reduce the overall cost of deployment. Accordingly, two users that express the same intent in terms of QoS and security level, will be serviced by the same SFC. This generates significant cost reductions, since a lower number of SFCs will be instantiated. However, this creates a new challenge of stressing the capabilities of the SFC that may result in a performance drift, since the queuing delay will exponentially increase with the load/number of intents waiting to be processed by a VNF [20].

Hence, when an SFC cannot receive any more intents of its type, the particular intent should be allocated to another SFC. We need to emphasize that the re-allocation of an intent to another SFC should not be perceived as a flow migration or actual flow relocation that could result in additional delays but a control decision of where a user's intent should be allocated from the beginning. Additionally, if an intent cannot be allocated to the requested SFC, the alternative SFC should still satisfy the requirements of the intent. For example, a "Medium QoS" intent should be placed to an SFC that offers higher than medium QoS (i.e., "High"). The same applies for the Security level.

C. Intent Relocation and Admission Control using Model Predictive Control (MPC)

In this section, we discuss how our approach tackles the problem defined in Section III-A. Specifically, two optimization problems are formulated, the solution of which retrieves the user relocation control strategy and the admission control strategy. The first one has a recurring nature, while the second one is invoked only when user relocations alone are not enough to guarantee SLA satisfaction and service has to be denied for certain users. To solve these optimization problems and deal with the identified constraints, we employ the MPC scheme twice; each time, it solves a linear optimization problem, which is based on the system model provided in the previous subsections, over a predefined time horizon. In particular, regarding the user relocation optimization, i.e., the first MPC invocation, at each control time slot t, the system behavior, $x_i(t)$, is observed and information, $\tilde{v}_i(t), \tilde{q}_i(t)$, is collected and used to update the dynamic model of the system, Eq. (1), for each SFC i. The variables $\tilde{v}_i(t), \tilde{q}_i(t)$ are predicted values of $v_i(t), q_i(t)$ respectively, the estimation of which is given in detail in the following subsection. Then, to accommodate the proactive nature of our solution, an optimization problem is solved over a prediction horizon of K time slots and the resulting control actions of the first time slot are applied to the system in a closed-loop control fashion, according to the receding horizon control principle [6]. An overview of this operation is depicted in Fig. 1.

Assuming that the act of relocating a user introduces additional labour for the infrastructure, we aim at achieving

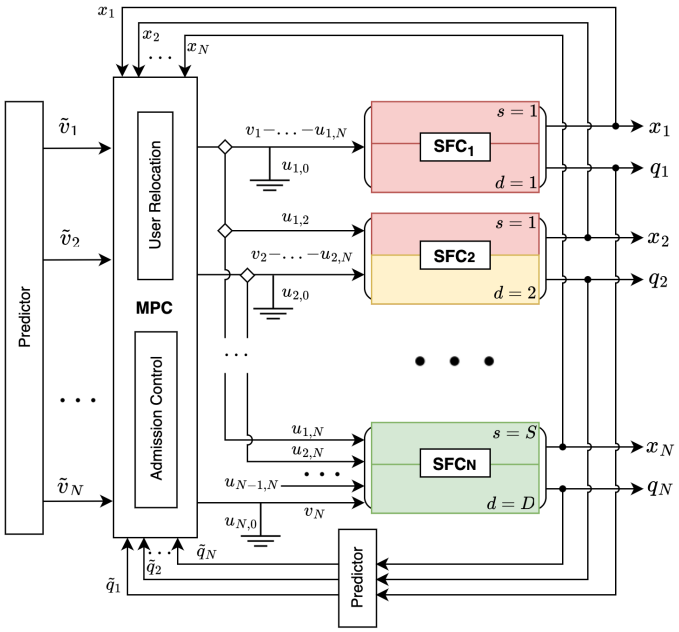


Fig. 1: MPC Framework Overview: Hierarchy of User Relocations formed by the assumed priorities.

the desirable SLA satisfaction while keeping the number of relocations in the control horizon to a minimum. Thus, the proactive user relocation optimization problem solved at each time slot t is formulated as:

$$\min_{u(k)} \quad \sum_{k=t+1}^{t+K} \frac{\delta}{\sqrt{k-t}} \cdot u(k) \tag{5a}$$

$$\text{s.t.} \quad u_{i,j}(k) \geq 0, \quad \forall u_{i,j}(k) \in u(k), \tag{5b}$$

$$0 \leq \sum\nolimits_{i,j \in [1,\dots,N]} u_{i,j}(k) \leq \tilde{v}_i(k), \quad u_{i,j}(k) \in u(k), \tag{5c}$$

$$x_i(k) \leq \mathcal{X}_i^{SLA}, \quad \forall i \in [1,\dots,N], \tag{5d}$$

$$x_i(k+1) = x_i(k) + b_i u(k) + c_i(\tilde{v}_i(k) - \tilde{q}_i(k)), \quad \forall i \in [1,\dots,N], \tag{5e}$$

where $\mathcal{X}_i^{SLA}$ is the desired maximum end-to-end delay and the coefficient $\delta = [\delta_{1,2}, \delta_{1,3}, \dots, \delta_{1,N}, \delta_{2,3}, \dots, \delta_{N-1,N}] \in \mathbb{R}_+^{1\times L}$ is a constant row vector which dictates the priority in which the user relocations take place, by assigning lighter or heavier weights to the corresponding control inputs. These priorities in our case reflect i) the preference to relocate the users to SFCs offering the same level of QoS with a higher level of security and, if that is infeasible, then to ii) relocate the users to the SFC offering the closest higher QoS level. Hence, the elements of δ are calculated as follows:

$$\delta_{i,j} = \begin{cases} w_1(d_j - d_i) + w_2(s_j - s_i), & \text{if } d_j - d_i \geq 0 \\ \gg 1, & \text{otherwise,} \end{cases}$$

where $w_1 \geq w_2, w_1, w_2 \in \mathbb{R}$ are empirically selected constants that produce $\delta_{i,j} \leq 1$. Relocations to SFCs with lower levels of QoS are discouraged, thus a very large weight $\delta_{i,j} \gg 1$ is assigned to the respective inputs. We also remind that relocations to SFCs with lower levels of security are prohibited

by design. Finally, relocations are discounted towards the end of the horizon where the prediction accuracy is lower.

When the first MPC fails to produce a solution, relocating the users alone is not sufficient to guarantee SLA satisfaction, thus the need for admission control arises. That is only when the second MPC invocation takes place; for this, we increment $u(t)$ with N new control variables, $u_{i,0}, i \in [1, ..., N]$, that correspond to the number of rejected users in each SFC. In this way, we define a new input vector

$$u'(t) = [u_{1,0}(t), u_{1,2}(t), ..., u_{2,0}(t), u_{2,3}(t), ..., u_{N-1,N}(t), u_{N,0}(t)]^\intercal \in \mathbb{Z}^{(L+N)\times 1}. \quad (6)$$

Consequently, a new row vector $b'_i \in \mathbb{R}^{1\times(L+N)}$ is calculated offline for each SFC by using the LSLCB algorithm, alongside a new incremented $\delta' \in \mathbb{R}_+^{1\times(L+N)}$, where $\delta'_{i,0} \gg \delta'_{i,j}$, $\forall i, j \in [1, ..., N]$. This condition ensures that rejecting users is not prioritized over relocating them. Then, the optimization problem (5) is solved with the new variables and coefficients in place and a control law consisting of both relocations and rejections of users is calculated. This double MPC solving procedure is guaranteed to provide a solution and is iterated for as long as the automatic control scheme is active.

D. User Flow Estimation

For each SFC i, the incoming number of users is estimated with the use of a Holt linear exponential smoothing filter [21] which captures the linear trend of time series. For any time interval t, the one-step prediction $\tilde{v}_i(t)$ of the incoming user flow $v_i(t)$ is:

$$\begin{aligned} \tilde{v}_i(t) &= \hat{v}_i(t) + \gamma(t), \\ \hat{v}_i(t) &= \alpha v_i(t) + (1-\alpha)(\hat{v}_i(t-1) + \gamma(t-1)), \\ \gamma(t) &= \beta(\hat{v}_i(t) - \hat{v}_i(t-1)) + (1-\beta)\gamma(t-1), \end{aligned} \quad (7)$$

where α and β are smoothing constants, $\hat{v}_i(t)$ is the smoothed value and $\gamma(t)$ denotes the linear trend in the measurements. As the initial values, a random $\hat{v}_i(0)$ is used, within the range of the incoming number of users and $\gamma(0) = 0.5$. Estimations for the outgoing number of users $\tilde{q}_i(t)$ are calculated in a similar way. To get predictions for a horizon deeper than one step, the prediction of the previous step is fed back as the real value, which naturally results in a decreasing accuracy.

Despite its popularity, this predictor tends to overestimate the real values [22], thus the control law calculated in problem (5) might not satisfy the input constraint (4) during the execution time. In this case, a control law $u''(t)$ is applied, which is calculated by arbitrarily decreasing input values $u_{i,j}$ in a round-robin fashion, until the constraint (4) is eventually satisfied. This results to an end-to-end delay $x''_i(t) \leq x_i(t) \leq \mathcal{X}_i^{SLA}, \forall i \in [1, ..., N]$, as the linear system (1) is monotone to the incoming number of users. On the other hand, when the real flows are underestimated, SLA violations might occur.

IV. Performance Evaluation

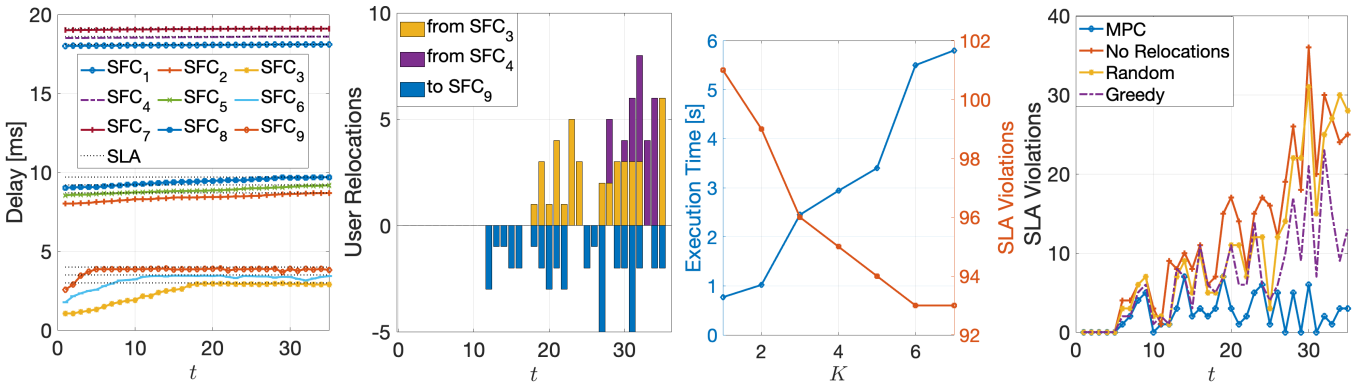
For the evaluation, a data center represented as a k=6 fat tree topology is considered. The propagation delay is set as $1ms$, $3ms$, and $5ms$ for the edge, aggregation and core links respectively [23]. Additionally, an *M/M/1* queuing delay model is followed as it can efficiently model the server queues [24]. The processing delay is a function of the VNF types used to form the SFCs and ranges from $100\mu sec$ to $400\mu sec$ [25].

Regarding the SFCs, a total of $N = 9$ services are considered that represent the different QoS ($D = 3$) and Security ($S = 3$) combinations presented in Section III. The bandwidth requirements of the SFC are set to 10 Mbps for "Low QoS" ($d_i = 1$), 30 Mbps for "Medium" ($d_i = 2$), and 50 Mbps for "High" ($d_i = 3$). For the modeling and parameter estimation part, the *MATLAB lsqlin* solver was used on a time-series dataset of $T = 2500$ samples. Each control time slot t spans for $10min$. The incoming intents v_i are issued according to a Poisson distribution with a rate of 4 intents per time slot, and their lifetime follows an exponential distribution with a range of $1 - 24$ time slots. The experiments last for a period of 35 time slots and the prediction horizon for the MPC solver is $K = 6$ time slots, unless stated otherwise.

As a first experiment, the performance of the MPC algorithm is evaluated Accordingly, Fig. 2a depicts the overall behavior of our system, which guarantees the SLA satisfaction of the SFCs, despite the fluctuating incoming user flows. For a better understanding, we zoom into the exact operation of three representative SFC placements, i.e. SFC_3 (Fig. 3a), SFC_6 (Fig. 3b) and SFC_9 (Fig. 3c). The system is assumed to have zero initial conditions, so its behaviour depends uniquely on the inputs. Here, we observe that when the experienced delay is close to the SLA maximum values (dotted line in Fig. 2a), incoming users start to be proactively switched from SFC_3 and SFC_6. Specifically, this response of our system begins at $t = 17$ for SFC_3 and at $t = 11$ for SFC_6, when the SLA value is reached for the first time and, from that point on, the incoming user flows are regulated with relocations, to keep the delay at an acceptable level. When the delay starts decreasing, the relocations are limited as well (e.g., $t = 23, 24, 28, 32$ for SFC_6). In the odd case of SFC_9, which by design and due to SLA constraints is unable to relocate any users, the invocation of the second MPC (admission control) results in incoming user rejections when the delay is close to the SLA value.

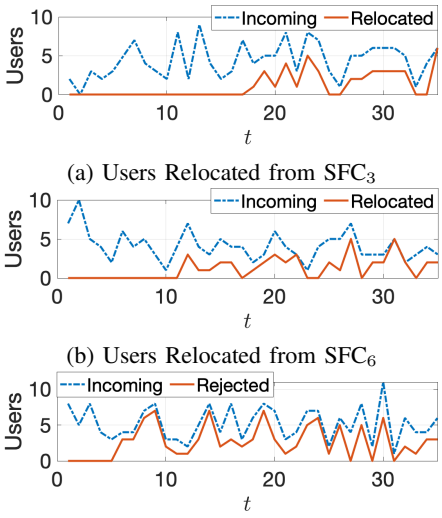
Exploring even further the behavior of our system, we concentrate on the user relocations that happen to and from SFC_6. In the breakdown depicted in Fig. 2b, we observe that when the SFC struggles, more users are relocated from it than to it (e.g., $t = 20, 22, 27$). On the other hand, when the delay decreases, it mostly receives users (e.g., $t = 23, 24, 32, 33$); in detail, SFC_6 is one of the relocation targets of SFC_3 and SFC_4, which proactively start relocating user flows to it when their end-to-end delay is close to the SLA value. We notice that the relocation priorities are respected as well.

Fig. 2c presents the effect of the MPC's prediction horizon K to its performance and execution time. The algorithm was implemented using the YALMIP toolbox [26]. As the proposed solution avoids any SLA violations, as shown in the analysis so far, we assume that only when denying an intent an SLA violation occurs. The average execution time is calculated per



(a) (b) (c) (d)

Fig. 2: MPC performance analysis: (a) SLA satisfaction, (b) Relocations breakdown (SFC_6), (c) Horizon length impact on execution time and service denials, (c) Benchmarking.



(a) Users Relocated from SFC_3

(b) Users Relocated from SFC_6

(c) Users Rejected at SFC_9

Fig. 3: System's response to fluctuating incoming user flows.

time slot and includes the invocation of the first and potentially the second MPC algorithm, depending on whether admission control is required in a slot. Apparently, as the length of the horizon K increases, the algorithm manages to calculate better proactive control strategies that result in less service denials (8% less in total, when moving from $K = 1$ to $K = 6$, for 1450 users). However, the performance remained the same for $K > 6$, indicating this is the sweet spot for our configuration. Naturally, increasing K results in increasing real execution times (up to 6 sec), however these are still negligible when compared to the length of the control slot t which spans minutes.

In the last part of the evaluation, we perform a comparison of the proposed algorithm, with three baseline solutions: i) a no-action one where no reallocations nor rejections of users take place, ii) one where a random percentage of incoming users are relocated from the SFCs that measure SLA violations, to other random ones and iii) a greedy solution, where the whole incoming user flow of struggling SFCs is relocated to the most underloaded ones, at each time slot t. To make a fair comparison, the same relocation priority constraints that were defined in Section III were implemented for all the algorithms, that is why service denials are still reported from SFCs that are not allowed to relocate their users (i.e., SFC_9).

Fig. 2d depicts the number of SLA violations of all algorithms per time slot. For the MPC, as in the previous analysis, the violations are a result of service denials; for the others, a summation of the rejected users and users experiencing end-to-end delays greater than their SLAs is made. The results showcase the dominance of the proposed solution, as not only the number of new SLA violations per slot are kept to a minimum compared to the baselines, but their trend of occurrence is static as well. In detail, in a total of 1450 incoming users throughout the duration of this experiment, the MPC reported a total of 6.4% of SLA violations, while the no-action one, the random and the greedy reported 30.3%, 24.9% and 17.1% respectively. This happens due to the increased ability of the MPC to deal with the anticipated workload and, based on the feedback from the SLA satisfaction, to proactively relocate users from overloaded SFCs to underloaded ones.

V. Conclusion

In this paper, we studied the problem of providing SLA guarantees for network assurance in IBN-enabled networks. A Model Predictive Control-based algorithm was introduced to proactively and optimally assign the incoming intents among the available SFCs. This was achieved by first modeling the operation of the said SFCs as LTI systems, offline, and then formulating the control problem of scheduling the relocations, as a mathematical programming problem. This problem is optimally solved in a receding horizon manner with the help of integrated predictors for the incoming and outgoing user flows. The experimentation results showed that the proposed algorithm can maximize the SLA satisfaction by successfully responding to the fluctuating workload of the system. Our future work lies on exploring the alternatives for dealing with extreme prediction inaccuracies of our system and incorporate them to the formal definition of the problem solved.

REFERENCES

[1] X. Zheng and A. Leivadeas, "Network Assurance in Intent-Based Networking Data Centers with Machine Learning Techniques," in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021, pp. 14–20.

[2] A. Clemm, L. Ciavaglia, L. Granville, and J. Tantsura, "Intent-Based Networking - Concepts and Definitions," Intent Research Task Force (IRTF), Tech. Rep., 2021.

[3] A. Leivadeas and M. Falkner, "A Survey on Intent Based Networking," *IEEE Communications Surveys & Tutorials*, p. 1, 2022.

[4] ——, "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.

[5] A. Laghrissi and T. Taleb, "A Survey on the Placement of Virtual Resources and Virtual Network Functions," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1409–1434, 2019.

[6] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[7] M. Medvetskyi, M. Beshley, and M. Klymash, "A Quality of Experience Management Method For Intent-Based Software-Defined Networks," in *2021 IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, 2021, pp. 59–62.

[8] N. Herbaut, C. Correa, J. Robin, and R. Mazo, "SDN Intent-based conformance checking: application to security policies," in *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*, 2021, pp. 181–185.

[9] R. Riggio, I. G. Ben Yahia, S. Latré, and T. Rasheed, "Scylla: A language for virtual network functions orchestration in enterprise WLANs," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 401–409.

[10] A. Mehmood, A. Muhammad, T. Ahmed Khan, J. J. Diaz Rivera, J. Iqbal, I. Ul Islam, and W.-C. Song, "Energy-efficient auto-scaling of virtualized network function instances based on resource execution pattern," *Computers & Electrical Engineering*, vol. 88, p. 106814, 2020.

[11] L. Velasco, S. Barzegar, D. Sequeira, A. Ferrari, N. Costa, V. Curri, J. Pedro, A. Napoli, and M. Ruiz, "Autonomous and Energy Efficient Lightpath Operation Based on Digital Subcarrier Multiplexing," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 9, pp. 2864–2877, 2021.

[12] D. Sanvito, D. Moro, M. Gullì, I. Filippini, A. Capone, and A. Campanella, "ONOS Intent Monitor and Reroute service: enabling plug&play routing logic," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 272–276.

[13] N. F. S. de Sousa, N. Islam, D. A. L. Perez, and C. E. Rothenberg, "Policy-Driven Network Traffic Rerouting Through Intent-Based Control Loops," *Anais do Workshop de Gerência e Operação de Redes e Serviços*, 2019.

[14] T. A. Khan, A. Mehmood, J. J. Diaz Rivera, and W.-C. Song, "Machine Learning Approach for Automatic Configuration and Management of 5G Platforms," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2019, pp. 1–6.

[15] K. Abbas, M. Afaq, T. A. Khan, A. Mehmood, and W.-C. Song, "IBNSlicing: Intent-Based Network Slicing Framework for 5G Networks using Deep Learning," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2020, pp. 19–24.

[16] S. Barzegar, M. Ruiz, and L. Velasco, "Reinforcement Learning - based Autonomous Multilayer Network Operation," in *2020 European Conference on Optical Communications (ECOC)*, 2020, pp. 1–4.

[17] M. Avgeris, D. Dechouniotis, N. Athanasopoulos, and S. Papavassiliou, "Adaptive resource allocation for computation offloading: A control-theoretic approach," *ACM Transactions on Internet Technology (TOIT)*, vol. 19, no. 2, pp. 1–20, 2019.

[18] R. J. Hanson, "Linear least squares with bounds and linear constraints," *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, pp. 826–834, 1986.

[19] A. Leivadeas, G. Kesidis, M. Falkner, and I. Lambadaris, "A Graph Partitioning Game Theoretical Approach for the VNF Service Chaining Problem," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 890–903, 2017.

[20] N. Pitaev, M. Falkner, A. Leivadeas, and I. Lambadaris, "Multi-VNF performance characterization for virtualized network functions," in *2017 IEEE Conference on Network Softwarization (NetSoft)*, 2017, pp. 1–5.

[21] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting methods and applications*. John wiley & sons, 2008.

[22] J. W. Taylor, "Exponential smoothing with a damped multiplicative trend," *International journal of Forecasting*, vol. 19, no. 4, pp. 715–725, 2003.

[23] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, 2015, pp. 171–177.

[24] D. Bhamare, M. Samaka, A. Erbad, R. Jain, L. Gupta, and H. A. Chan, "Optimal virtual network function placement in multi-cloud service function chaining architecture," *Computer Communications*, vol. 102, pp. 1–16, 2017.

[25] A. Leivadeas, M. Falkner, and N. Pitaev, "Analyzing Service Chaining of Virtualized Network Functions with SR-IOV," in *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*, 2020, pp. 1–6.

[26] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*. IEEE, 2004, pp. 284–289.