# Network Assurance via Hybrid Systems modelling and Set Invariance

Marios Avgeris

*Informatics Institute, Faculty of Science, University of Amsterdam, Science Park 900, Amsterdam, 1098 XH, The Netherlands*

Nikolaos Athanasopoulos

*School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, 16 Malone Rd, Belfast, BT9 5BN, United Kingdom*

Aris Leivadeas

*Dept. of Software and Information Technology Engineering, École de Technologie Supérieure, University of Quebec, 1100 R. Notre Dame O, Montréal, QC H3C 1K3, Canada*

Ioannis Lambadaris

*Dept. of Systems and Computer Engineering, Carleton University, 3336 Mackenzie Building, 1125 Colonel By Dr, Ottawa, ON K1S 5B6, Canada*

**Abstract**

Network Function Virtualization (NFV) along with Service Function Chaining (SFC) associate network functions with virtual resources to offer end-to-end network services. The goal is to enhance the operator's environment with elasticity, agility, and cost-effectiveness, automating the allocation of services while guaranteeing fulfilment of their specification, i.e., achieving the *Network Assurance* (NA). We address the challenges of meeting Quality of Service (QoS) specifications, such as delay and security, in fluctuating traffic and a multi-tenant SFC infrastructure. First, we model the evolution of the average time delays as a piecewise affine dynamical system. By exploiting the monotonicity properties of the system dynamics, we characterise and compute the operating regions where feasible scheduling strategies guaranteeing NA exist, by calculating the maximal controlled invariant set of the system. We propose a Model Predictive Control (MPC) framework leading to recursively feasible real-time decision strategies, thus, guaranteeing NA. We present an extensive comparison with commonly used approaches from the realms of heuristics and Machine Learning and show our framework outperforms them, guaranteeing at the same time *a priori* all specifications.

*Keywords:* Service Function Chaining, Model Predictive Control, Set invariance, Network Assurance.

## 1. Introduction

The dawn of 6G and beyond networking era has seen the proliferation of end devices (e.g., wearables, the Internet of Things - IoT) and resource-hungry applications (e.g., AR/VR, AI), resulting in major transformations in network architectures. The Edge-Cloud continuum becomes extremely relevant in this setting, allowing distribution of computational and storage resources, reducing latency and enhancing reliability towards meeting stringent requirements [1]. On the one hand, this expansion in infrastructure size and complexity has significantly increased the needs for automation, with the traditional manual configuration of the network being no longer sustainable. On the other hand, the increased QoS and service reliability requirements necessitate formal performance guarantees: the infrastructure should adjust to any performance drift that may result to Service Layer Agreement (SLA) violations, and performance degradation must be promptly detected and addressed [2].

By virtualizing network services and functions, service requests can be carried out by a Service Function Chain (SFC) composed of a number of Virtualized Network Functions (VNFs), such as Firewalls, Intrusion detection/prevention systems (IDS/IPS) and Access Control Lists (ACLs) [3][4]. SFCs can share network functionalities among different users [5]. However, managing concurrent, diverse, and competing service requests, e.g., in terms of delay constraints and security specifications, is challenging. In this context, *network assurance (NA)* refers to the capability of guaranteeing that the QoS and operational constraints of deployed services remain satisfied throughout runtime operation, despite dynamically changing workloads and service conditions [6][7][8]. This requires continuously monitoring and controlling the network behavior in order to detect and mitigate deviations from the desired service performance in real time. Merely solving the problem of service request-to-SFC assignment only guarantees that the performance is satisfied upon instantiating the requested service, while the network state can dynamically change throughout its lifetime leading to performance uncertainty [9]. Therefore, there is a need for methods that combine computational tractability with formal guarantees for network assurance in realistic SFC deployments.

To tackle these challenges, we propose a new SFC-enabled network assurance model in the form of a piecewise affine discrete time dynamical system with inputs. In this model, the end-to-end (E2E) delays are captured explicitly as state variables. Building on [9], which focused on the Intent-Based Networking (IBN) use case without providing NA guarantees, we extend it to the generic SFC-oriented networking model. In the context of this work, NA is interpreted as the existence of admissible scheduling strategies that guarantee confinement of the closed-loop system trajectories within predefined QoS-admissible operating regions for all admissible workload and service-rate realizations. Specifically, we use Model Predictive Control (MPC) [10] to formulate the NA problem as a finite optimisation problem which we subsequently solve in real time. To ensure QoS compliance at all times, we characterise the set of admissible states and corresponding scheduling strategies that ensure confinement of the trajectories within an appropriately defined constraint set. The main contributions of this paper can be summarised as follows:

- *Dynamic Models for Network Assurance:* We consider an SFC-enabled infrastructure handling service requests with heterogeneous requirements, modeled as flow rates. We provide a new model of the infrastructure, namely, a positive piecewise affine system. The incoming request and service rates are the exogenous signals of the system, while the states capture the SFC QoS performance in terms of E2E delays. The control inputs are the distribution of the request rates to the SFCs. This model defines QoS specifications explicitly as state and input constraints, together with the profile of the incoming request rates, service rates, and SFC structure.
- *Controlled Invariant Sets characterisation and computation*: The obtained model in the general case is computationally hard to analyse. By identifying monotonicity properties of the system dynamics and properties of the shape of the related *input-dependent* constraint set induced by the problem specifications, we introduce a new method for computing efficiently the maximal controlled invariant set. This set constitutes the maximal subset of the space of E2E service delays where the closed-loop trajectory can be confined through suitable, admissible scheduling strategies, thereby guaranteeing NA and persistent QoS satisfaction under admissible workload and service-rate variations.
- *Guaranteed NA*: We address the associated scheduling problem using MPC. We pose as optimisation constraints the trajectory containment in the precomputed controlled invariant sets, and introduce penalty terms in the optimisation cost that correspond to infrastructure costs. Importantly, we relax the assumptions on the variability of the workload and service rate profiles. To address scalability, we break down the system into hierarchically connected subsystems and solve the respective MPC problems in cascading order.
- *Comparison with the state-of-the-art* (SoTA): The performance is evaluated through extensive simulations using real data from various representative scenarios against three SoTA works, namely, greedy, semi-greedy, and Reinforcement Learning (RL)-based. Our framework outperforms the other approaches in QoS satisfaction, system cost reduction under fluctuating volumes of incoming request rates and constraint satisfaction.

To the best of our knowledge, this is the first work introducing a control-theoretic framework that enables formally guaranteed network assurance under realistic system dynamics. We model the dynamic characteristics commonly observed in the Edge-Cloud continuum, taking explicitly into account uncertain and fluctuating service rates and incoming flows. Through designing and enforcing robust control strategies, our

approach achieves guarantees on Key Performance Indicators (KPIs), such as service times, and any metrics modelled as state variables within specified safe sets. As such, our framework can continuously ensure that network responses meet the required specifications even in the face of unforeseen disturbances. This aligns with the need for real-time assurance in next generation network service quality (e.g., latency, reliability) and security [11]. We emphasise that formal guarantees on constraint satisfaction and performance provide a comparative advantage over conventional alternatives such as scalable learning-based methods. Our approach compares favorably, partially because these methods do not offer guarantees while they focus on scalability for scheduling in larger setups. Thus, our motivation is to fill the gap between scalable but heuristic approaches and the lack of formal approaches that are computationally efficient for a relatively small, however realistic for NA, number of parameters. The remainder of the paper is structured as follows. Section 2 highlights the related work. Section 3 presents the system model and the problem statement. Section 4 outlines the controlled invariant sets characterisation and computation. In Section 5 the control scheme is presented. The results are demonstrated in Section 6. Finally, Section 7 concludes our work and provides possible future directions.

## 2. Related Work

### *2.1. Network Assurance*

Research in NA and real-time control that meets user requirements is gaining attention, e.g., the Zero-Touch Management (ZTM) in 5G and 6G networks [12]; specifically, automated service assurance is positioned among the prominent automation capabilities that improve deployment agility, reduce costs, and ensure reliable network performance without human intervention. ZTM architectures, as defined by the European Telecommunications Standards Institute (ETSI), enable end-to-end service lifecycle automation through intent-driven closed-loop control mechanisms that coordinate monitoring, analysis, and actuation across multiple domains [13]. An effort to standardize network assurance is within the Closed Control Loop SLS Assurance (COSLA) framework, defined in 3GPP TS 28.535 [14]. COSLA is a management control loop for NA that consists of monitoring, analytics, decision, and execution. A closed control loop for the communication service is deployed in the preparation phase and takes effect during the operation phase of the lifecycle management, to adjust the resources used for the network services. Soldani *et al.* in [15], propose the extended Berkeley Packet Filter (eBPF) as a practical enabler for cloud-native observability, networking, and security, particularly suited for 5G and beyond networks. The authors suggest that their framework is suitable for instrumentation that enables near real-time performance monitoring and assurance. In the IBN context, a proactive mechanism is presented in [16], where Perepu *et al.* discussed the application of multi-agent reinforcement learning (MARL) to achieve intent-based service assurance in cellular networks. The approach prioritised satisfaction of key performance indicators such as Quality of Experience, based on availability of resources. However, as shown in the results, this learning-based approach failed to provide strict formal guarantees for the intents' requirements. Bridging IBN with ZTM, Khan *et al.* introduce machine learning to support proactive orchestration and optimisation of network slicing and resource management, for example through traffic prediction and adaptive routing strategies [17]. Similarly, Abbas *et al.* integrate AI-driven analytics to enable intelligent zero-touch service provisioning and lifecycle management in heterogeneous environments [18]. While these approaches demonstrate the effectiveness of automation and optimisation, they primarily focus on performance-driven objectives and lack formal guarantees on system-level behaviour and assurance. In contrast, our work focuses on incorporating such guarantees within an automated, intent-driven framework.

### *2.2. Service Function Chaining Assurance*

Typical specifications in SFC-enabled NA are reliability, performance, QoS and security, with works focusing on continuous monitoring, dynamic resource allocation, and real-time optimization. Xie *et al.* [19] focus on machine learning analytics and propose a framework that integrates monitoring and orchestration tools. The CHain Installation, Monitoring and Adjustment (CHIMA) [20] framework is designed for the deployment and performance assurance of SFCs. Similar to our approach, this is achieved by rerouting

traffic flows in case of requirement violations. Security requirements however are not treated. Nouruzi *et al.* [21] propose a Deep Reinforcement Learning (DRL)-based framework for online service provisioning in NFV-enabled networks, setting an optimization problem with constraints on QoS and network resource limitations. NA is addressed by introducing a penalty in the reward function when QoS requirements are not met; in this case the request is rejected. However, there are no formal guarantees in place for requirement satisfaction during the lifetime of the service request. In the same direction, Lei *et al.* in [22] introduce a method for deploying QoS-assured multi-tenant SFCs using stochastic prediction models for VNF latency. This approach predicts the latency distribution using random-forest regression in order to minimise end-to-end latency and bandwidth consumption, however without considering security requirements or performance after the requests are deployed.

### *2.3. Piecewise Affine Modeling & MPC Applications*

Recent advances in hybrid and switched-system control have also explored computationally efficient and less conservative optimization frameworks for complex dynamical systems. For instance, You *et al.* have proposed switching-type MPC schemes for Takagi–Sugeno fuzzy systems to improve controller flexibility and reduce conservatism by dynamically adapting the control strategy according to changes in the system dynamics [23]. In parallel, recent work on networked dynamical systems by Gao *et al.* has investigated computationally efficient control formulations that avoid the complexity growth associated with lifted optimization approaches while preserving desirable convergence properties [24]. These directions further highlight the growing interest in scalable and adaptive control-theoretic techniques for complex interconnected systems. Building on this broader trend, our work investigates network assurance in SFC-enabled infrastructures through a monotone piecewise affine formulation combined with invariant-set-based MPC scheduling.

## 3. System Model

We consider service request rates with requirements that vary among $k$ levels of security and $n$ levels of QoS for a requested network service respectively. Each multi-tenant SFC realizes a unique combination of QoS and security level which leads to $k \times n = d$ different SFC placements. Security level is achieved through the VNF structure, ranging from basic features (i.e., a virtual router followed by a firewall) to advanced ones (Intrusion Detection System - IDS and IP encryption function - IPSec). QoS level depends on the placement of said VNFs. Towards achieving proactive QoS-assurance, we follow the work in [25] where time is considered slotted. To avoid SLA violations, service requests should either be assigned to an SFC offering the requested or a higher-level combination of security and QoS levels, as long as the average experienced E2E delay remains within the SFC's QoS tolerance. This design aligns with the specifications of RFC 7665 [26], where traffic is evaluated based on specific criteria and directed into the appropriate Service Function Paths (SFPs) to ensure consistent service delivery.

The set of nonnegative reals is $\mathbb{R}^n_+$. Vector inequalities hold componentwise, and the vector with all components equal to one is 1. For a matrix $A \in \mathbb{R}^{n\times m}$, its $ij$-th element is $(A)_{ij}$. The Hadamard product between two matrices $A \in \mathbb{R}^{n\times m}, B \in \mathbb{R}^{n\times m}$ is $A \odot B \in \mathbb{R}^{n\times m}$, with $(A \odot B)_{ij} = (A)_{ij}(B)_{ij}$, $i = 1, .., n$, $j = 1, ..., m$. The vectorisation operation of a matrix $A \in \mathbb{R}^{n\times m}$ is $\mathrm{vec}(A) \in \mathbb{R}^{nm}$. For a vector $x \in \mathbb{R}^d$, the maximum operation with zero $y = \max\{x, 0\}$ holds componentwise, i.e, $y_i = \max\{x_i, 0\}$. We consider the vector spaces $X = \mathbb{R}^d$ and $Y = \mathbb{R}^m$. The orthogonal projection of a set $\mathcal{S} \subset X \times Y$ to a subspace $X$ is

$$\pi(\mathcal{S}, X) = \{x \in X : (\exists y \in Y : [x^\top \quad y^\top]^\top \in \mathcal{S})\}.$$

### *3.1. Dynamics*

In any given time period, the system processes an integer number of incoming requests and completes an integer number of requests. To effectively optimise the system's performance, we consider a periodic scheduling. Thus, within each sampling interval, incoming and outgoing requests are treated as continuous flows. This modelling approach allows for a granular treatment of traffic dynamics and is often used in

resource allocation and scheduling problem formulations [27]. Thus, for every unit of time we assume an incoming request rate for each SFC $v \in \mathbb{R}^d_+$, and an outflow of completed requests, i.e., service rate $q \in \mathbb{R}^d_+$. As states, we consider the average end-to-end delays $x \in \mathbb{R}^d_+$, experienced over a unit of time at each SFC. The control inputs distribute the requests to the appropriate SFCs. We organise these inputs in a square matrix $U \in \mathbb{R}^{d\times d}$. Specifically, the element $(U)_{ij}$ corresponds to the allocation of the incoming request rate $i$ to the SFC $j$. The system can thus be written as

$$x(t+1) = f(x(t), U(t), q(t)), \tag{1}$$

with

$$f(x, U, q) = \max\{\bar{f}(x, U, q), 0\},$$

and

$$\bar{f}(x, U, q) = a \odot x + (B \odot U)1 - e \odot q, \tag{2}$$

where $a, e$ are nonnegative vectors, $a \in \mathbb{R}^d_+$, $e \in \mathbb{R}^d_+$, and $B \in \mathbb{R}^{d\times d}_+$ encodes the admissible request-to-SFC allocation structure. We remind that 1 denotes a vector of ones of compatible dimension. The dynamics (2) concerns the evolution of the end-to-end time delays $x$ that depend on the current value of the delay (first term $a \odot x$), the weighted sum of added request flow (second term $(B \odot U)1$), and the subtracted serviced requests (third term $-e \odot q$). The dynamics (1) is the componentwise maximum with zero, restricting the time delays to be nonnegative, effectively rendering the positive orthant invariant for $x$. Consequently, the resulting system (1) is a positive piecewise affine system.

**Example 1.** *The SLA specifications, such as constraints on request rate distribution, are embedded in the model via the structure of the input matrix $B$. For example, we often consider a hierarchical structure where the $i^{th}$ request rate can be assigned to the $i^{th}$ SFC, or to a higher one, namely, $j^{th}$ SFC, with $i < j \leq d$, consequently, we have $\frac{d(d+1)}{2}$ control inputs. To enforce the structure illustrated in Figure 1, the inputs are organised in a lower triangular matrix $U \in \mathbb{R}^{d\times d}$, with*

$$U = \begin{bmatrix} u_{11} & 0 & \cdots & 0 \\ u_{21} & u_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ u_{d1} & u_{d2} & \cdots & u_{dd} \end{bmatrix}.$$

If $x_i(t+1)$ depends also on the other delays $x_j(t)$ additional to $x_i(t)$, for example, when the services are hosted on the same infrastructure the term $a \odot x$ in (1) can be replaced by $Ax$, for some matrix $A \in \mathbb{R}^{d\times d}$. System (1) shares some conceptual similarities with a Vector Addition System (VAS) [28] in terms of the linear additive nature of state transitions while ensuring non-negativity. However, in our case the states are real compared to integers in VAS. System (1) is not linear due to the max operator involved, while (2) is linear dynamics. The choice of system (1) is made to enforce nonnegativity of the state variables, i.e.,

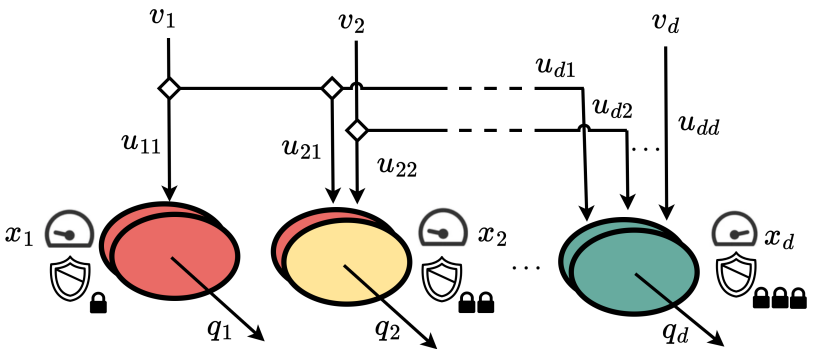



Figure 1: Example 1 - System Model Overview.

the end-to-end delay vector $x$. We note we choose this structure over other possible modelling approaches, such as bilinear systems, or Linear Parameter Varying systems (LPVs), e.g., [29]. Such representations involve products between inputs $U$ and the exogenous signal of incoming requests, with the inputs being the percentage of the distribution of incoming request rates. Such choices inevitably induce complexities hard to analyse [30]. To estimate the parameters of system (1), one can employ standard identification methods such as Constrained Least Squares [31], once and offline on data acquired from the SFCs' operation.

### *3.2. Constraints*

We consider the vector of time-varying incoming request rate $v \in \mathbb{R}^d$ and set $V = \mathbb{R}^d$. Similarly, we consider the vector of service rate $q \in \mathbb{R}^d$, and correspondingly the set $Q = \mathbb{R}^d$. We slightly abuse notation, and by $\text{vec}(U) \in \mathbb{R}^{\hat{d}}$, $\hat{d} \leq d^2$, we denote the vectorized form of the inputs, i.e., the nonzero elements of the matrix $U$. In the $\text{vec}(U) - V - Q$ space we define constraints describing the range where the request rates and service rates lie, effectively inducing constraints on the range of $U$. In a real-world network, incoming and outgoing traffic are interdependent, possibly in a non-linear way. Such constraints can model physical limitations like finite bandwidth and processing capacities, security and policy constraints and traffic prioritization [32]. We adopt a realistic assumption that these constraints are convex and can thus be approximated efficiently by compact polytopic sets located in the positive orthant. Consequently, setting $d^* = \hat{d} + 2d$, we define $\mathcal{W} \subseteq \mathbb{R}_+^{d^*}$,

$$\mathcal{W} = \left\{ \begin{bmatrix} \text{vec}(U) \\ v \\ q \end{bmatrix} \in \mathbb{R}_+^{d^*} : G_u \,\text{vec}(U) + G_v v + G_q q \leq w \right\}, \tag{3}$$

with $G_u \in \mathbb{R}^{p \times \hat{d}}$, $G_v \in \mathbb{R}^{p \times d}$, $G_q \in \mathbb{R}^{p \times d}$, $w \in \mathbb{R}^p$. Given any vector $[v^\top \;\; q^\top]^\top \in \pi(\mathcal{W}, V \times Q)$, we consider the set $\mathcal{W}(v, q) \subset \mathbb{R}^{\hat{d}}$ defined as

$$\mathcal{W}(v, q) = \{\text{vec}(U) \in \mathbb{R}^{\hat{d}} : G_u \,\text{vec}(U) \leq w - G_v v - G_q q\}.$$

### *3.3. Problem statement*

**Definition 1** ([33]). *A set $\mathcal{S} \subset \mathbb{R}_+^d$ is controlled invariant with respect to the system* (1) *and the constraints* $\left[\text{vec}(U)^\top \;\; v^\top \;\; q^\top\right]^\top \in \mathcal{W}$ (3) *if for any $x \in \mathcal{S}$, and any $[v^\top \;\; q^\top]^\top \in \pi(\mathcal{W}, V \times Q)$, there exists an input $\text{vec}(U) \in \mathcal{W}(v, q)$ such that $f(x, U, q) \in \mathcal{S}$. The controlled invariant set $\mathcal{S}_M$ is the maximal controlled invariant set (MCIS) if for any other controlled invariant set $\mathcal{S}$ it holds $\mathcal{S} \subseteq \mathcal{S}_M$.*

Characterising a set with controlled invariance allows for an *a priori* guarantee of QoS performance, i.e., *network assurance*, for the studied problem; if the initial conditions $x \in \mathbb{R}_+^d$ lie in a controlled invariant set $\mathcal{S}$, then for any possible combination of request rate and service rate, there exists at least one scheduling strategy for these requests to the deployed SFCs so that the QoS requirements are always satisfied. The QoS requirements of the infrastructure are encoded through the admissible state and input constraints defining acceptable operating regions for the SFC delays and scheduling decisions. Consequently, achieving network assurance is equivalent to guaranteeing that the closed-loop trajectories of system (1) remain within these admissible regions for all admissible workload and service-rate realizations.

**Problem 1.** *Consider the system* (1) *and the set $\mathcal{W} \subset \mathbb{R}_+^{d^*}$* (3)*. Compute the MCIS.*

Controlled invariance characterises the set of states for which an admissible control strategy confining the trajectories in the set exist. A natural question is how to develop such controllers. In a networking infrastructure, SFCs fulfilling different levels of service requests are optimised for different types of traffic. Reassigning traffic to non-optimised SFCs, although potentially beneficial for matching requests requirements, can lead to inefficient use of network resources.

**Problem 2.** *Consider the system* (1) *and the set $\mathcal{W} \subset \mathbb{R}_+^{d^*}$* (3)*, and a controlled invariant set for the system* (1)*. Develop a scheduling strategy implementable in real-time such that NA holds.*

Problem 1 is well known in set-based methods [33] and formal methods [34], with established algorithms and related convergence results for important families of systems. To the best of our knowledge, the system under study (1), although important for scheduling applications, has not been considered. There are two challenges: First, (1) is not linear, thus, immediate application of standard techniques in [33] is not possible. Second, the coupled constraint set $\mathcal{W}$ does not allow elimination of the quantifier *for all* $q, v$, thus, the elimination operations of erosion and projection are not possible and straightforward respectively [35]. We tackle both challenges in Section 4. To solve Problem 2, we define an appropriate MPC scheme, whose recursive feasibility is possible by the construction of the invariant set. To propose a scalable solution, instead of using a single MPC for the whole system, we design local controllers for each one of the $k$ subsystems that decide the distribution of incoming request rates among the group of their respective $n$ SFCs. The controllers are hierarchically connected to each other, allowing the migration of requests between subsystems when deemed necessary. The design of this control scheme is detailed in Section 5.

## 4. Controlled invariant set computation

To guarantee Network Assurance, we first establish a methodology for computing the maximal controlled invariant set for the system under study, thus, characterising the subset of the state space where admissible controllers guaranteeing NA can be found. Our approach aligns with reachability-based algorithms that are based on iterative computations of the backward reachable map [33], or preimage map

$$\mathcal{C}(\mathcal{S}) = \{x \in \mathbb{R}^d : (\forall (v,q) \in \pi(\mathcal{W}, V \times Q), \exists U \in \mathcal{W}(v,q) : f(x,U,q) \in \mathcal{S})\}. \tag{4}$$

Computing (4) is not straightforward, because the dynamics (1) is not linear. Instead, we consider the map

$$\bar{\mathcal{C}}(\mathcal{S}) = \{x \in \mathbb{R}^d : (\forall (v,q) \in \pi(\mathcal{W}, V \times Q), \exists U \in \mathcal{W}(v,q) : \bar{f}(x,U,q) \in \mathcal{S})\}. \tag{5}$$

The map $\bar{f}(x,U,q)$ is linear. However, the fact that the exogenous signals $q, v$ are coupled with the inputs does not allow straightforward computations of the preimage map (5). In what follows, we first establish that it is sufficient to compute the backward reachability map considering only the vertices of the set $\pi(\mathcal{W}, V \times Q)$, denoted by $[v_i^\top \ q_i^\top]^\top$, $i = 1, ..., p$. We note that the number of vertices of $\pi(\mathcal{W}, V \times Q)$ is finite as it is the orthogonal projection of a polytopic set [36]. We consider the maps

$$\bar{\mathcal{C}}_i(\mathcal{S}) = \{x \in \mathbb{R}^d : (\exists U \in \mathcal{W}(v_i, q_i) : \bar{f}(x,U,q_i) \in \mathcal{S})\}. \tag{6}$$

**Proposition 1.** *Consider the system* (1)*, the dynamics* (2)*, the set* $\mathcal{W}$*, and let* $[v_i^\top \ q_i^\top]^\top$*,* $i = 1, .., p$ *be the vertices of the set* $\pi(\mathcal{W}, V \times Q)$*. For any compact polytopic set* $\mathcal{S} \subset \mathbb{R}^d_+$*, it holds that*

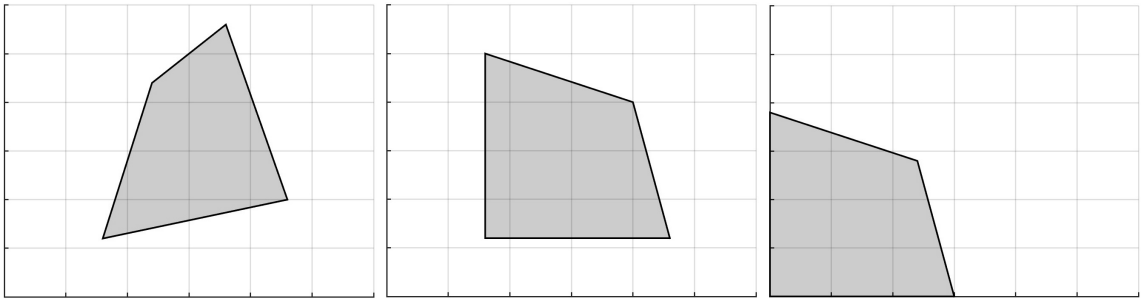
$$\bar{\mathcal{C}}(\mathcal{S}) = \bigcap_{i=1}^{p} \bar{\mathcal{C}}_i(\mathcal{S}). \tag{7}$$

*Proof.* By Carathéodory's theorem, see, e.g., [37, Theorem 17.1], for any $[v^\top \ q^\top]^\top \in \pi(\mathcal{W}, V \times Q) \subset \mathbb{R}^{2d}$, there are at most $2d + 1$ vertices such that

$$\begin{bmatrix} v \\ q \end{bmatrix} = \sum_{i=1}^{2d+1} \lambda_i \begin{bmatrix} v_i \\ q_i \end{bmatrix}, \tag{8}$$

with $\sum_{i=1}^{2d+1} \lambda_i = 1, \lambda_i \geq 0$. Let any vector $x \in \bigcap_{i=1}^{2d+1} \bar{\mathcal{C}}_i(\mathcal{S})$. Consider inputs $U_i \in \mathcal{W}(v_i, q_i)$ such that $\bar{f}(x, U_i, q_i) \in \mathcal{S}$. Such inputs exist by definition of $\bar{\mathcal{C}}_i(\mathcal{S})$. Let $U = \sum_{i=1}^{p} \lambda_i U_i$, with $\lambda$ satisfying (8). Then, by linearity of the dynamics (2) we have

$$\bar{f}(x,U,q) = \bar{f}\left(x, \sum_{i=1}^{2d+1} \lambda_i U_i, \sum_{i=1}^{2d+1} \lambda_i q_i\right) = \sum_{i=1}^{2d+1} \lambda_i \bar{f}(x, U_i, q_i).$$



(a) $\mathcal{S} \notin \mathbb{M}$ (b) $\mathcal{S} \in \mathbb{M}$ (c) $\mathcal{S} \in \mathbb{M}_0$

Figure 2: Example 2 - Illustration of class $\mathbb{M}$, $\mathbb{M}_0$ of sets.

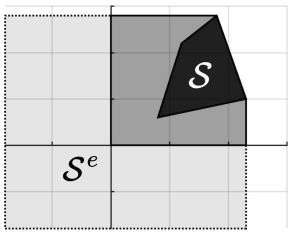



Figure 3: Example 2 - $\mathcal{S} \notin \mathbb{M}$ (black), $\mathcal{S}^e \in \mathbb{M}$ (light grey), and $\mathcal{S}^e \cap \mathbb{R}^d_+ \in \mathbb{M}_0$ (dark grey). It holds $\mathcal{S} \subset \mathcal{S}^e \cap \mathbb{R}^d_+ \subset \mathcal{S}^e$.

For any $i = 1, ..., p$, it holds that $\bar{f}(x, U_i, q_i) \in \mathcal{S}$, thus, $\bar{f}(x, U, q) \in \mathcal{S}$ by convexity of the set $\mathcal{S}$. Moreover, since $U_i \in \mathcal{W}(v_i, q_i)$ it holds that $G_u \operatorname{vec}(U_i) + G_v v_i + G_q q_i \leq w$, and for any nonnegative $\lambda_i$ it follows $G_u \operatorname{vec}(\lambda_i U_i) + G_v \lambda_i v_i + G_q \lambda_i q_i \leq \lambda_i w$. Summing the previous linear inequalities for $i = 1, .., p$ it follows that

$$G_u \operatorname{vec}(U) + G_q q + G_v v \leq w,$$

consequently, $U \in \mathcal{W}(v, q)$. Thus, for any $x \in \cap_{i=1}^p \bar{\mathcal{C}}_i(\mathcal{S})$ and for any $[v^\top \ q^\top]^\top \in \pi(\mathcal{W}, V \times Q)$ there is an admissible input $U \in \mathcal{W}(v, q)$ such that $\bar{f}(x, U, q) \in \mathcal{S}$, or, $\cap_{i=1}^p \bar{\mathcal{C}}_i(\mathcal{S}) \subseteq \bar{\mathcal{C}}(\mathcal{S})$. To show equality, suppose there is $x \in \bar{\mathcal{C}}(\mathcal{S}) \setminus \cap_{i=1}^p \bar{\mathcal{C}}_i(\mathcal{S})$, which is a contradiction because there is an index $i \in [1, p]$ so that there does not exist $U \in \mathcal{W}(v_i, q_i)$ so that $\bar{f}(x, U, q_i, v_i) \in \mathcal{S}$, and consequently, $x \notin \bar{\mathcal{C}}(\mathcal{S})$. □

To associate the mapping $\bar{\mathcal{C}}(\mathcal{S})$ with the actual map $\mathcal{C}(\mathcal{S})$ we wish to compute, i.e., the backward reachability map of (1), we introduce a particular family of sets $\mathcal{S}$ as follows.

**Definition 2.** *A convex set $\mathcal{S} \subset \mathbb{R}^d$ belongs to the class $\mathbb{M}$ if for any two vectors $x, y \in \mathbb{R}^d$ that belong to the set $\mathcal{S}$, any vector $z \in \mathbb{R}^d$ satisfying $x \leq z \leq y$ belongs in the set $\mathcal{S}$ as well. Additionally, if $\{0\} \subset \mathcal{S}$, then we say the set $\mathcal{S}$ belongs to class $\mathbb{M}_0$, or, $\mathcal{S} \in \mathbb{M}_0$.*

We next define the extension of a set $\mathcal{S}$.

**Definition 3.** *For any set $\mathcal{S} \subset \mathbb{R}^d$, we define its extension $\mathcal{S}^e \subset \mathbb{R}^d$ (where 'e' stands for 'extended') as follows*

$$\mathcal{S}^e = \{x \in \mathbb{R}^d : (\exists y \in \mathcal{S} : x \leq y)\}. \tag{9}$$

**Example 2.** *In Figure 2 we depict three examples of sets with respect to classes $\mathbb{M}$, $\mathbb{M}_0$. In Figure 3, the extension operator is visualized. It should be clear for any set $\mathcal{S} \subset \mathbb{R}^d_+$ it holds $\mathcal{S}^e \cap \mathbb{R}^d_+ \in \mathbb{M}_0$.*

We present the second result that leads to computations recovering the backward reachable set for (1).

**Proposition 2.** *Consider a set $\mathcal{S} \in \mathbb{M}_0$. It holds*

$$\mathcal{C}(\mathcal{S}) = \bar{\mathcal{C}}(\mathcal{S}^e). \tag{10}$$

*Proof.* First, we show that $x \in \bar{\mathcal{C}}(\mathcal{S}^e)$ implies $x \in \mathcal{C}(\mathcal{S})$, i.e., $\bar{\mathcal{C}}(\mathcal{S}^e) \subseteq \mathcal{C}(\mathcal{S})$. Consider any point $x \in \bar{\mathcal{C}}(\mathcal{S}^e)$. By definition, for any $[v^\top \ q^\top]^\top \in \pi(\mathcal{W}, Q \times V)$, there is an input $U \in \mathcal{W}(v, q)$ such that $\bar{f}(x, U, q) \in \mathcal{S}^e$. If $\bar{f}(x, U, q) \in \mathcal{S} \subseteq \mathcal{S}^e$ then necessarily $x \in \mathcal{C}(\mathcal{S})$ because in this case $\bar{f}(x, U, q) = f(x, U, q)$. If $\bar{f}(x, U, q) \notin \mathcal{S}$, then necessarily there is at least one component $k \in [1, d]$ so that $\bar{f}_k(x, U, q) < 0$, which in turn implies $f_k(x, U, q) = 0$. By assumption, since $\mathcal{S} \in \mathbb{M}_0$, it follows that $f(x, U, q) \in \mathcal{S}$. Thus, $\bar{\mathcal{C}}(\mathcal{S}^e) \subseteq \mathcal{C}(\mathcal{S})$. Next, we show that $x \in \mathcal{C}(\mathcal{S})$ implies $x \in \bar{\mathcal{C}}(\mathcal{S}^e)$, i.e., $\mathcal{C}(\mathcal{S}) \subseteq \bar{\mathcal{C}}(\mathcal{S}^e)$. By hypothesis $\mathcal{S} \in \mathbb{M}_0$, thus if $f_i(x, U, q) = \max\{0, \bar{f}_i(x, U, q)\} = 0$, necessarily $\bar{f}_i(x, U, q) \leq 0$, which immediately implies $\bar{f}(x, U, q) \in \mathcal{S}^e$, or, $x \in \bar{\mathcal{C}}(\mathcal{S}^e)$. The proof is complete. □

**Proposition 3.** *Consider a polytopic set $\mathcal{S} \in \mathbb{M}_0$. Then, the set $\mathcal{C}(\mathcal{S})$ is a polytopic set, and furthermore $\mathcal{C}(\mathcal{S}) \in \mathbb{M}_0$.*

*Proof.* The first part of the statement is straightforward from Proposition 2, Proposition 1 and the fact that (2) is linear. To show that $\mathcal{C}(\mathcal{S}) \in \mathbb{M}_0$, we first consider each element $\bar{\mathcal{C}}_i(\mathcal{S}^e)$ (6) in the intersection (7), and show that $\bar{\mathcal{C}}_i(\mathcal{S}^e) \in \mathbb{M}_0$. Indeed, for any $x \in \bar{\mathcal{C}}_i(\mathcal{S}^e)$, suppose there is some $y \leq x$ such that $y \notin \bar{\mathcal{C}}_i(\mathcal{S})$. Let $\beta \geq 0$ such that $x = y + \beta$. There is $U \in \mathcal{W}(v_i, q_i)$ such that $\bar{f}(x, U, q_i) \in \mathcal{S}^e$. Then, $\bar{f}(y, U, q_i) = \bar{f}(x - \beta, U, q_i) = \bar{f}(x, U, q_i) - a \odot \beta \leq \bar{f}(x, U, q_i)$, thus, $\bar{f}(y, U, q_i) \in \mathcal{S}^e$. Consequently, $\bar{\mathcal{C}}_i(\mathcal{S}^e) \in \mathbb{M}_0$. To show the same for the intersection, i.e., that $\cap_{i=1}^p \bar{\mathcal{C}}_i(\mathcal{S}^e) \in \mathbb{M}_0$ it is sufficient to show that $\mathcal{S}_A \cap \mathcal{S}_B \in \mathbb{M}_0$ for any two sets $\mathcal{S}_A \in \mathbb{M}_0$, $\mathcal{S}_B \in \mathbb{M}_0$. First, observe that $\{0\}$ is in both sets $\mathcal{S}_A, \mathcal{S}_B$. Next, take any $x \in \mathcal{S}_A \cap \mathcal{S}_B$, $y \in \mathcal{S}_A \cap \mathcal{S}_B$ such that $x \leq y$. For any $z$ satisfying $x \leq z \leq y$, it holds that $z \in \mathcal{S}_A$, $z \in \mathcal{S}_B$, thus, $z \in \mathcal{S}_A \cap \mathcal{S}_B$, and the set $\mathcal{S}_A \cap \mathcal{S}_B \in \mathbb{M}_0$. Since this relation holds for any intersection of a finite number of sets, $\bar{\mathcal{C}}(\mathcal{S}) \in \mathbb{M}_0$, and by (10), $\mathcal{C}(\mathcal{S}) \in \mathbb{M}_0$. □

**Theorem 1.** *Consider the system (1), the dynamics (2), the set $\mathcal{W} \subset \mathbb{R}^{d^*}$, and let $[v_i^\top \ q_i^\top]^\top$, $i = 1, .., p$ be the vertices of the set $\pi(\mathcal{W}, V \times Q)$. Consider the set sequence $\{\mathcal{S}_l\}$, where*

$$\mathcal{S}_{l+1} = \left( \bigcap_{i=1}^{p} \bar{\mathcal{C}}_i(\mathcal{S}_l^e) \right) \cap \mathcal{S}_0, \tag{11}$$

*where $\mathcal{S}_0$ is a compact polytopic set and $\mathcal{S}_0 \in \mathbb{M}_0$. Then, the set sequence $\{\mathcal{S}_l\}$ converges to the MCIS.*

*Proof.* From Propositions 1, 2, 3 it follows that for any $l \geq 0$, $\left(\bigcap_{i=1}^p \bar{\mathcal{C}}_i(\mathcal{S}_l^e)\right) = \bar{\mathcal{C}}(\mathcal{S}_l^e) = \mathcal{C}(\mathcal{S}_l)$. Consequently, the set recursion (11) is equivalent to $\mathcal{S}_{l+1} = \mathcal{C}(\mathcal{S}_l) \cap \mathcal{S}_0$. Thus, $\{\mathcal{S}_l\}$ is the sequence of backward reachable sets for the system (1). The set sequence $\{\mathcal{S}_l\}$ is a nested monotonically decreasing sequence of compact polytopic sets of class $\mathbb{M}_0$, and convergent in the space of compact sets equipped with Hausdorff distance metric to a (possibly empty) compact set. Following standard argumentation as in [33, Chapter 5] it follows that the fixed point is the maximal controlled invariant set with respect to the system (1). □

**Example 3.** *In Figure 4 we depict an example of the MCIS computation for a 3-SFC networking infrastructure modelled as a third order system. As the initial set $\mathcal{S}_0$, we consider the polytope formed by the ranges of acceptable end-to-end delays for the three involved SFCs, i.e., $x_i \in [0, x_{i,\max}]$, $i = 1, 2, 3$, which is illustrated as the outer cube in white. The MCIS is retrieved after the convergence of the set sequence (11) and is depicted with the dark grey colour. The system parameters are*

$$a = \begin{bmatrix} 1.02 \\ 1 \\ 1 \end{bmatrix}, B = \begin{bmatrix} 0.10 & 0 & 0 \\ 0.07 & 0.05 & 0 \\ 0.39 & 0.09 & 0.24 \end{bmatrix}, e = \begin{bmatrix} 0.10 \\ 0.05 \\ 0.24 \end{bmatrix}.$$

*Convergence is achieved in 96 iterations, and the resulting set is described by 36 vertices. For higher-order systems, the computation time and the complexity of the representation of the MCIS grow significantly for*

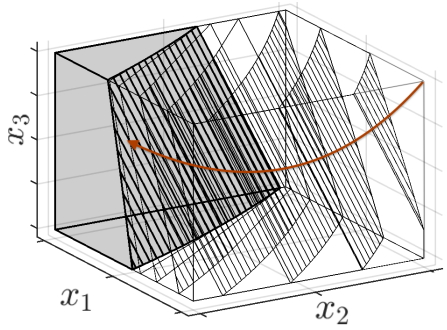



Figure 4: Example 3 - MCIS computation for a 3-SFC networking infrastructure modelled as a third order system.

*specific choices of constraint sets* (3) *and dynamics* (1). *This is expected, since the computation involves iterative polyhedral operations whose complexity depends not only on the state dimension, but also on the number of facets/vertices generated during the backward-reachability recursion. In our implementation, the computation of an MCIS for a typical 3-SFC subsystem is in minutes, while for a 4-SFC subsystem the computation time increases significantly. We emphasize that this computation is performed offline during the controller design phase, whereas the online controller only uses the precomputed invariant sets as constraints in the MPC problem. These observations further motivate the computation-aware control architecture presented in Section 5, where the global problem is decomposed into smaller subsystem-level MPC problems. Improving the scalability of the offline invariant-set computation, for example through relaxed invariance notions, approximate polyhedral representations, or decomposition-based set computations, is an important direction for future work.*

## 5. Control for Guaranteed Network Assurance

Controlled invariant sets characterise the selection set of controllers that render the set invariant. Herein, we propose a receding horizon control scheme that guarantees NA while minimising a desired performance metric. As discussed in Section 3, scheduling the service request rates among the SFCs introduces operational and communication overhead. Additionally, the lower the end-to-end service delay, the higher the QoS satisfaction. To this purpose, we consider a control horizon $N \geq 1$, and the cost function

$$J\left(x_0, \{U_i\}_{i=0}^{N-1}\right) = \sum_{l=0}^{N-1} c_l(x_l, U_l) + c_N(x_N), \tag{12}$$

where $c_l(x_l, U_l) = \|(C_l \odot U_l)1\|_1 + \beta_l \|x_l\|_1, \quad l = 1, .., N-1$ is the horizon step, and $c_N(x_N) = \beta_N \|x_N\|_1$, with $C_l \in \mathbb{R}_+^{d \times d}$, $\beta_l \in \mathbb{R}_+^d$. The weights $C_l, \beta_l$ define the priority in which the request rate distribution takes place, by assigning lighter or heavier weights to the corresponding control inputs. These priorities in our case reflect a sequential preference order based on the difference between the offered QoS level by the involved SFCs. We remind that typically distribution to SFCs with lower levels of QoS and/or security is prohibited by design.

**Example 4.** *We consider the hierarchical system setup in Example 1, consisting of d SFCs. One possible selection of values is*

$$C_l = \alpha \begin{bmatrix} 0 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 2 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ d-1 & d-2 & \dots & 0 \end{bmatrix}, \qquad \beta_l = \beta 1, \quad l = 1, ..., N,$$

*for scalars* $\alpha > 0, \beta > 0$. *We make the following assumptions.*

**Assumption 1.** *At time instant t the workload and service rates $v(t+l), q(t+l)$ respectively, $l = 0, ..., N-1$, are available up to time $t + N - 1$.*

Assumption 1 is a realistic assumption in our context, based on the practical necessity for operators to analyse network traffic and predict short-term future traffic to proactively manage services and systems. Solutions range from time-series modelling [38] to machine learning-based predictors [39] with high accuracy.

**Assumption 2.** *At each time instant t, incoming request and service rate signals are consistent with $\mathcal{W}$ (3), i.e., $[v(t)^\top q(t)^\top]^\top \in \pi(\mathcal{W}, V \times Q)$, $t \geq 0$.*

Given the system (1), the constraints set $\mathcal{W}$, and the controlled invariant set $\mathcal{M}$, the problem to be solved at time $t$ is

$$\min_{\{U_i\}_{i=0}^{N-1}} J(x_0, \{U_i\}_{i=0}^{N-1}) \tag{13}$$

$$\text{s.t.} \quad x_0 = x(t), \tag{14}$$

$$x_{l+1} = f(x_l, U_l, q_l), \quad l = 0, 1..., N-1, \tag{15}$$

$$x_l \in \mathcal{M}, \quad l = 1, ..., N, \tag{16}$$

$$U_l \in \mathcal{W}(v_l, q_l) \quad l = 0, ..., N-1, \tag{17}$$

where $q_l = q(t + l), v_l = v(t + l)$, $l = 0, ..., N - 1$.

**Proposition 4.** *Consider the system (1), a controlled invariant set $\mathcal{M} \subset \mathbb{R}^d_+$, the control law $U(t) = U_0^*$, where $\{U_i^*\}_{i=0}^{N-1}$ is the optimizer of the problem (13)–(17), and let $x(t_0) \in \mathcal{M}$. Then, for all $t \geq t_0$ the constraint set (14)–(17) is nonempty, and $x(t) \in \mathcal{M}$.*

*Proof.* Since $x(t_0) \in \mathcal{M}$, under Assumption 2 there is $U(t_0)$ so that $f(x(t_0), U(t_0), q(t_0) \in \mathcal{M}$. Consequently, the constraint (15) is satisfied for $l = 0$ and the constraint (16) is satisfied for $l = 1$. By applying the same reasoning, there is $U(l)$, $l = 0, ..., N - 1$ so that the relationships can be satisfied until $l = N - 1$ (15) and $l = N$ (16). Moreover, by definition of controlled invariance, $U(t_0) \in \mathcal{W}(v(t_0), q(t_0))$ and since the constraint set (17) covers $\mathcal{W}(v, q)$ for any admissible choice of $(v(t_0), q(t_0) \in \pi(\mathcal{W}, V \times Q)$, the input $U(t)$ satisfies these constraints, for any $t \geq t_0$. Consequently, $x(t_0 + 1) = f(x(t_0), U_0^*, q(t_0)) \in \mathcal{M}$. Since the system (1) is time invariant, the above statement holds for all $t \geq t_0$. □

### *5.1. Excess workload offloading in practical settings*

Assumption 2 is not always reasonable in dynamic settings; bursts of workload can occur for a service due to unexpected demand (e.g., a high publicity event), network operation anomalies (e.g., device outages and configuration changes) and network abuse anomalies (e.g., cyberattacks) among others [40], so that temporarily the request workload/service rate violates the constraints. One way to address this challenge would be to change the admissible constraint set $\pi(\mathcal{W}, V \times Q)$ so that it allows this outlier configuration of $(v(t), q(t))$. However, such a choice will inevitably lead to smaller, or even empty controlled invariant sets. Thus, to avoid imposing stringent constraints that are seldom active, we allow, at the worst case, the *offloading* of excess workload. This offloading can take the form of rejecting the excess workload or temporary migrating it to a system with enough serving capacity. To this purpose, we define the cost function

$$J_o\left(x_0, \{U_i\}_{i=0}^{N-1}\right) = \sum_{l=0}^{N-1} c_{o,l}(x_l, U_l) + c_N(x_N), \tag{18}$$

with

$$c_{o,l}(x_l, U_l) = \|(C_l \odot U_l)1\|_1 + \|\beta_l \odot x_l\|_1 + \gamma\|v_l - U_l 1\|_1,$$

$l = 1, .., N-1$, for some large value of $\gamma$, and by relaxing the constraint within the constraint set $\mathcal{W}$ that corresponds to the distribution of incoming requests from $u_l = U_l 1$, $l = 0..., N-1$, to

$$v_l \geq U_l 1, \quad l = 0, ..., N-1. \tag{19}$$

We note the results of Proposition 4 hold as well in this case. By appropriate choice of the weights $\gamma$ in the cost (18) we can penalise offloading, e.g., by setting $C_l = 0$, $\beta_l = 0$.

*5.2. Cascade MPC for scalable, information-restricted Network Assurance*

For efficient practical implementations, we restructure the system's centralized architecture into a hierarchical one consisting of $k$ subsystems. These subsystems comprise groups of SFCs ordered in terms of their offered security level; each subsystem consists of $n$ SFCs ordered in terms of their offered QoS level. This approach distributes the control computation load across subsystems, allowing the infrastructure to handle a larger number of requests simultaneously and efficiently; this prevents bottlenecks and ensures smoother operation even for larger SFC-enabled networks. Moreover, the proposed architecture improves privacy by limiting information dissipation, it localises better potential points of failure and reduces the risk of a single point of attack; if one subsystem fails, the others can continue to operate, ensuring continuous service availability [41]. In this case, the offloading process is realized as workload *migration* between the different subsystems. We consider subsystems of the type

$$x^i(t+1) = \max\{\bar{f}_i\left(x^i(t), U^i(t), q^i(t)\right), 0\}, i = 1, ..., k,$$

where

$$\bar{f}_i(x^i, U^i, q^i) = a^i \odot x^i + B^i \odot U^i - e^i \odot q^i. \tag{20}$$

We note that since the excess workload is migrated from a system to the next one, the performance of subsystem $i$ depends on the choice of the inputs $U^j$ of the subsystems $j = 1, ..., i$ and the workload rates $v^j$, $j = 1, ..., i$. The incoming request rates of subsystem $i$ becomes $\hat{v}^i = \sum_{j=1}^{i} v^j - \sum_{j=1}^{i-1} U^j 1$.

**Example 5.** *Figure 5 shows these interconnections; the $k-th$ subsystem implies $k-th$ level of security. For example, at the k-th level, it is allowed to carry out service requests that originally required a security level $1, ..., k-1$.*

The suggested MPC scheme distributes the incoming service requests among the available SFCs so that each closed loop subsystem trajectory of the states $x^i(t)$, $t \geq 0$, $i = 1, .., k$, is confined in its respective controlled invariant sets and satisfies the subsystem QoS. The MPC optimization problem is solved over a prediction horizon of $N$ units of time and the resulting control actions of the first unit of time are applied to each subsystem in a closed-loop control fashion, according to the receding horizon control principle [10]. In detail, we preserve the formulation in (18)-(19) for the subsystem cost as in the hierarchically connected subsystems we translate the offloading process to migrating the undistributed request rates of subsystem $i$ to subsystem $i+1$; the actual offloading might happen only in subsystem $k$, which in this case can account for a costly but resourceful deployment, e.g., the cloud. This emerging constrained MPC problem solved at unit of time $t$ for each local controller and a time horizon $N$ is

$$\min_{\{U_l\}_{l=0}^{N-1}} J_o\left(x_0^i, \{U_l\}_{l=0}^{N-1}\right) \tag{21}$$

$$\begin{aligned}
\text{s.t.} \quad & x_0^i = x^i(t), && (22)\\
& x_{l+1}^i = f(x_l^i, U_l^i, q_l^i), \quad l = 0, 1..., N-1, && (23)\\
& x_l^i \in \mathcal{M}^i, \quad l = 1, ..., N && (24)\\
& U_l^i \in \mathcal{W}^i(\hat{v}_l, q_l^i), \quad l = 0, ..., N-1, && (25)\\
& \hat{v}_l^i = \sum_{j=1}^{i} v_l^j - \sum_{j=1}^{i-1} U_l^j 1 = U_l^i 1, \quad l = 0, ..., N-1, && (26)
\end{aligned}$$

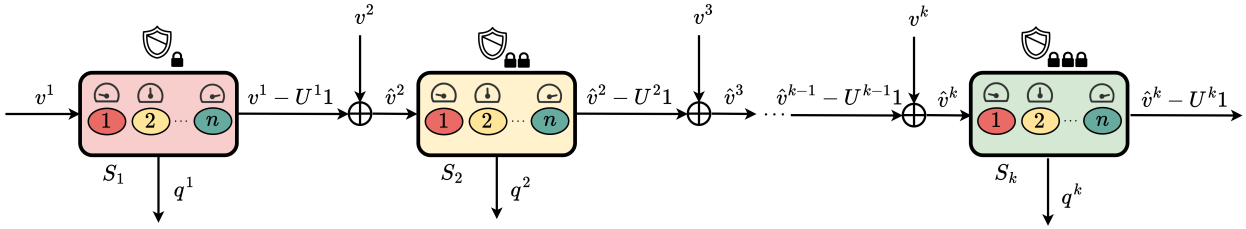


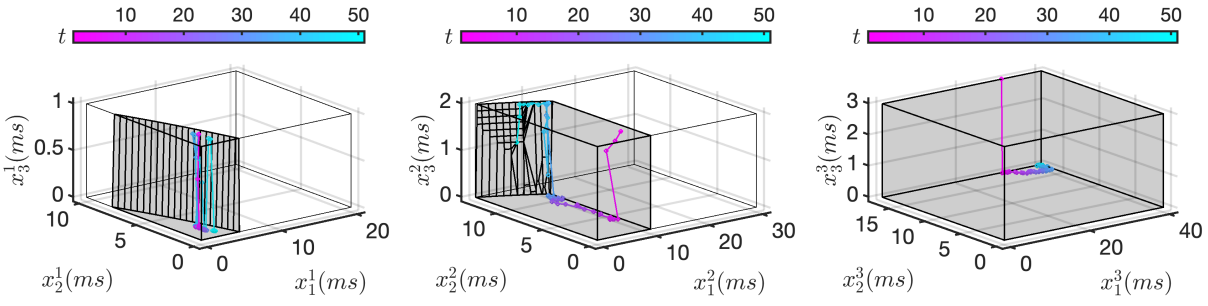

Figure 5: Example 5 - The structure of the hierarchical system with dynamics (20). The inputs to each subsystem, apart from the first $S_1$ depends on the previous inputs and the incoming request rates, i.e., $\hat{v}^i = \sum_{j=1}^{i} v^j - \sum_{j=1}^{i-1} U^j 1$.

where $q_l^i = q^i(t+l), v_l^i = v^i(t+1)$, $l = 0, ..., N-1$. We emphasize that the above optimization problem needs to be solved $k$ times, sequentially. In fact the optimiser $U^{i*}$ of MPC problem $i$, is given as input to the MPC problem $i+1$, as depicted in Figure 5. In this way, for every MPC $i$, the optimisation variables are $U_l^i$, $i = 0, ..., N-1$. We also need to note that the $\mathcal{M}^i$ controlled invariant sets are computed independently as the constraint sets $\mathcal{W}^i$ are decoupled.

## 6. Performance Evaluation

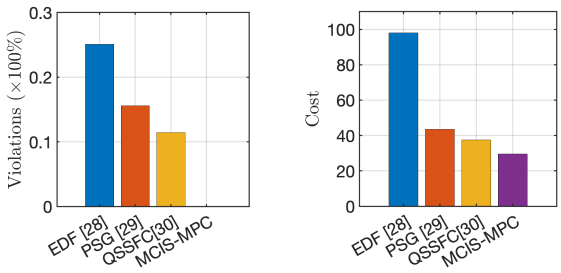
### *6.1. Simulation setup*

We simulate an SFC-enabled infrastructure with a data center in a 6 fat tree topology. The infrastructure is organized in $k = 3$ subsystems of different security levels, where each one contains $n = 3$ SFCs (offering increasing levels of QoS), realizing $d = 9$ QoS and security service level combinations, as presented in Section 5.2. The input structure in Example 1 defines the constraints on the distribution of the incoming request rates within each subsystem. The matrices defining the constraint set $\mathcal{W}$ (3) are in Appendix A. For parameter estimation, `MATLAB`'s `lsqlin` solver was used on a dataset of 2500 samples (computation time $\sim 80ms$), compiled using a proprietary discrete-event simulator provided by an industrial partner [42] to generate the request traffic (Poisson-exponential lifetime distribution) and has been validated in real testbeds [43]. The parameters for the 3 identified subsystems are in Appendix A. The experimentation runs for a period of 50 time slots. The MPC prediction horizon is set to $N = 5$ time slots. The influence of priorities in the numerical evaluation is twofold: First, the structure of the admissible input matrix $U$, defined through the matrix $B$, constrains the allowable request-to-SFC assignments according to the QoS and security hierarchy. Second, the weighting matrices and penalty terms in the MPC cost function $J_o$ (18), instantiated following Example 4, prioritise specific scheduling and migration decisions during runtime optimization. To define the MPC cost $J_o$, we follow the same Example with $\alpha = 1, \beta = 1, \gamma = 20$. The cloud-deployed SFC $(3, 3)$ acts as a contingency option; to discourage the reliance on the cloud deployment,




(a) MCIS and constraints ($S_1$). (b) MCIS and constraints ($S_2$). (c) MCIS and constraints ($S_3$).

Figure 6: Evolution of the system in time: the local MPCs select the admissible control inputs in real time so that the closed loop subsystem trajectories are confined in their respective MCIS.




(a) QoS Violations. (b) System Cost ($J_o$).

Figure 7: Overall performance comparative analysis.

we assign a larger coefficient $\gamma = 30$ to the migrations towards it. The QoS SLA thresholds $x_{i,max}$ for each of the $i = 1, ..., 9$ SFCs are given in Table 1. Optimization problems are solved using the `Gurobi Optimizer`. The experiments were performed in `MATLAB R2024b` running on a MacBook Pro, 11-core M3 Pro processor, 18 GB RAM.

To allow for practical implementation, service requests need to be handled as discrete entities. This necessitates the conversion of the continuous, real-number input values, resulted by solving the $k$ MPC problems (21)-(26) into integer values. To do so, we implement the optimal service rates using the deterministic weighted round-robin algorithm.

### *6.2. Network assurance through MCIS for MPC*

First, we demonstrate the performance of the cascading MPCs scheme. We calculate the MCIS $\mathcal{M}^i$, $i = 1, 2, 3$, for each of the three subsystems, depicted in Figure 6. In detail, $\mathcal{M}^1$ is described by 48 vertices, $\mathcal{M}^2$ by 105 and $\mathcal{M}^3$ by 8 vertices. The sets of acceptable end-to-end delays are also outlined as the outer cubes. We observe that $\mathcal{M}^3$ is the whole acceptable set induced by the QoS specifications; this is a result of SFC $(3, 3)$ being deployed to the cloud and having an increased service rate. We generate traffic using the simulator, solving the cascading MPC problems in each subsystem, and measuring the average end-to-end delays of the SFCs. The initial states are ${x_0}^1 = [2.6\ 1.9\ 1]$, ${x_0}^2 = [2.7\ 2\ 2]$ and ${x_0}^3 = [30\ 17\ 3]$. The trajectory of each subsystem is shown in Figure 6. In the comparative analysis that follows, further insight is provided on the efficacy of the proposed scheme.

### *6.3. Comparison*

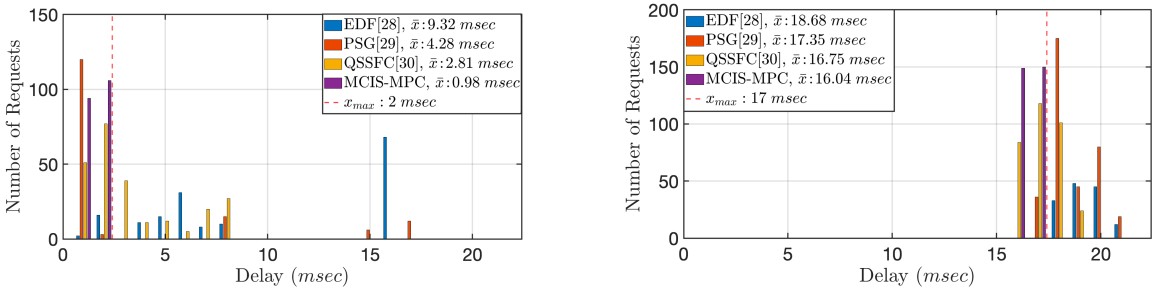
We denote our method as *MCIS-MPC* and compare against the following representative works from the literature:

1) *Earliest Deadline First (EDF)* [44]: EDF is a delay-aware scheduling algorithm which gives higher priority to requests with a lower deadline, i.e., highest QoS requirements. For fairness of comparison, we modify EDF in the following two ways: i) valid request distribution follows the rules defined in the simulation setup and ii) the SFC selection phase follows a greedy approach, in the sense that priority is given to the SFCs with the lowest non-SLA violating end-to-end delay at decision time.

2) *Priority-aware Semi-Greedy (PSG)* [45]: PSG semi-greedily optimises the total cost of the network during request distribution, while meeting the deadline, i.e., the QoS requirement, of the service requests.

| **QoS** / **Security** | **Low** (1) | **Medium** (2) | **High** (3) |
|---|---|---|---|
| **Low** (1) - $S_1$ | 20 | 10 | 1 |
| **Medium** (2) - $S_2$ | 30 | 13 | 2 |
| **High** (3) - $S_3$ | 40 | 17 | 3 |

Table 1: SFCs QoS SLA thresholds (milliseconds).




(a) Delay of requests carried out by SFC $(2, 3)$.

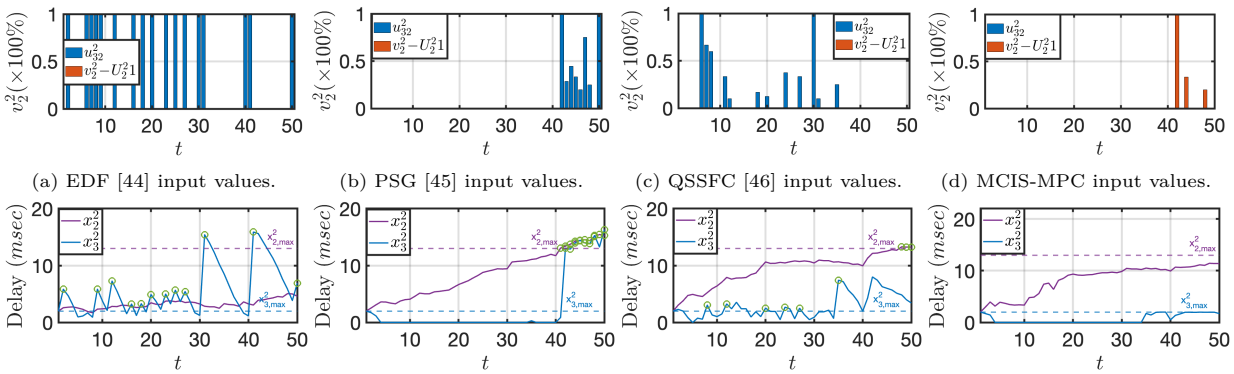
(b) Delay of requests carried out by SFC $(3, 2)$.

Figure 8: Comparing the histogram distributions of service requests' end-to-end delays, for two representative SFCs.

If the deadline of a given request is not met, it is allocated to the SFC resulting in the minimum violation. We modify it as follows: i) valid request distribution follows the rules defined in the simulation setup and ii) the energy consumption minimisation objective is replaced by the MPC's optimisation objective $J_o$.

3) *Q-learning-based Security SFC Selection (QSSFC)* [46]: QSSFC uses a modified $\epsilon$-greedy exploration strategy that picks out multiple candidate actions. We modify it as follows: i) the agent actions are constrained by the rules defined in the simulation setup and ii) for the reward, security benefit scores are mapped to security levels $k$, the cost is modified to reflect the MPC's optimisation objective $J_o$ and the impact on service quality accounts for the drift between the SFC's end-to-end delay $x$ and its SLA threshold $x_{max}$. We train the agent for 6000 episodes.

Assuming the same traffic and SFC states, we benchmark the overall performance of the four schemes in terms of QoS SLA violations and system cost ($J_o$). An SLA violation for an SFC corresponds to a service delay above the SLA threshold (Table 1). The results are demonstrated in Figure 7. In Figure 7a we observe that the EDF scheme reports the most SLA violations 25.11% since the incoming requests are distributed without consideration of the SLA thresholds. PSG performs better reporting 15.58% violations; this improvement is expected since this scheme considers the SLA threshold of the request and the current end-to-end delay of the targeted SFC before making the distribution decision, however, greedily treating the requests as they come. The QSSFC reports 11.44% violations, which shows that an RL method with a well-shaped reward function can respect the SLA threshold more efficiently. Nevertheless, the fact that the constraints are included in the form of penalty in the reward cannot alone guarantee strict adherence. EDF yields the highest system cost. Since the cost is incorporated in the decisions, we next observe a drop, with the QSSFC performing marginally better than PSG. Our solution manages to achieve the lowest system cost; that is 70% lower than EDF, 32% lower than PSG, and 21% lower than QSSFC.




(a) EDF [44] input values.

(b) PSG [45] input values.

(c) QSSFC [46] input values.

(d) MCIS-MPC input values.

(e) EDF [44] QoS violations.

(f) PSG [45] QoS violations.

(g) QSSFC [46] QoS violations.

(h) MCIS-MPC QoS violations.

Figure 9: Benchmarking the QoS impact of the request distribution and migration decisions at SFC $(2, 2)$.

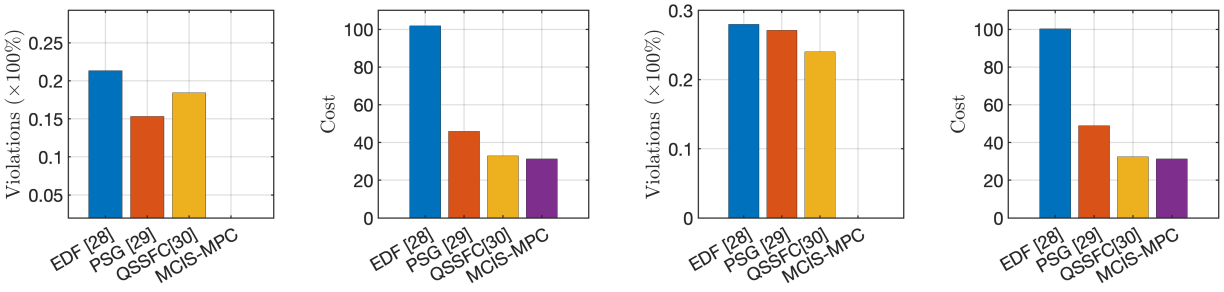
To get a more granular understanding of the performance of the MCIS-MPC mechanism, in Figure 8 we showcase the histogram distributions of the service requests end-to-end delays, for two representative SFCs for the compared schemes. For the medium-security, high-QoS SFC $(2,3)$ (Figure 8a), the EDF fails to satisfy the SLA for most of the requests, reporting an average of 9.32 $msec$ while the threshold is 2 $msec$. EDF also yields the widest disperse. PSG and QSSFC perform significantly better, with 4.28 $msec$ and 2.81 $msec$ average end-to-end delays respectively, which are still higher than the threshold. Their disperse is narrower as well. Notably, our method manages by construction to guarantee the SLA for all the requests, scoring a 0.98 $msec$ average delay with significantly less average dispersion. Similar results are in Figure 8b for the high-security, medium-QoS SFC $(3,2)$. We note the total served requests by the two SFCs vary between schemes, as their numbers are affected by the distribution and migration decisions.

Finally, we demonstrate the MCIS-MPC request distribution decisions by focusing on the medium-security, medium-QoS SFC $(2,2)$. Specifically, in Figure 9 we examine the input values as computed by each of the four compared schemes and the impact they have on the involved SFCs' QoS. We remind that based on the assumed structure of the input matrix, the service requests $v_2$ for SFC $(2,2)$ can be carried out by itself (input $u^2_{22}$), by the medium-security, high-QoS SFC $(2,3)$ (input $u^2_{32}$), or can be migrated to the high-security subsystem $S_3$ (input $v^2_2 - U^2_2 1$). In Figs. 9a-9d we omit $u^2_{22}$ since it comprises the majority of the incoming requests; the rest of the inputs are represented as percentages of $v^2_2$. In Figs. 9e-9h we depict the end-to-end delays of SFC $(2,2)$ $(x^2_2)$ and SFC $(2,3)$ $(x^2_3)$, together with their SLA thresholds (with dashed lines). We specifically highlight with green circles the QoS violations that occur after a request distribution/migration decision that involves the corresponding SFC.

In Figs. 9a and 9e, we observe that when $x^2_3 < x^2_2$ the EDF scheme greedily distributes all the incoming requests to SFC $(2,3)$. This leads to unnecessarily low end-to-end delays for SFC $(2,2)$ and a high number of QoS violations for SFC $(2,3)$. The semi-greedy decision making of the PSG scheme brings an improvement over EDF, which manages to keep both SFCs delays lower than the threshold, by carrying out all the incoming $v^2_2$ requests at SFC $(2,2)$, until $t = 41$. Then SFC $(2,2)$ starts reporting violations and thus part of $v^2_2$ is distributed to SFC $(2,3)$ resulting in a high number of violations for both (Figs. 9b and 9f). We note PSG selects SFC $(2,3)$ over migrating the requests to the high-security subsystem $S_3$, since the absolute value of the end-to-end delay is still comparatively lower at SFC $(2,3)$. On the other hand, the RL-based agent of the QSSFC scheme, manages a more balanced distribution of the requests between the two SFCs, as seen in Figure 9c, which results in fewer QoS violations (Figure 9g). Still, significant violations appear (e.g., at $t = 35$). The superiority of the proposed MCIS-MPC scheme is once again evident; its distribution decisions confine the trajectories of the end-to-end delays of both SFCs not only within their SLA thresholds, but within the even stricter MCIS of the subsystem (Figure 9h). In detail, every incoming $v^2_2$ request is carried out at SFC $(2,2)$ until it reaches the MCIS threshold, at $t = 41$ (Figure 9d). Then, requests are partially migrated to $S_3$, as SFC $(2,3)$ is also operating at its SLA threshold and thus it is not a viable alternative. We note that we omit the MCIS threshold on Figure 9h for simplifying the presentation.

### *6.4. Robustness under varying workload conditions*

To further investigate the robustness of the proposed framework and its sensitivity to parameter variations, we evaluated the compared approaches under two additional workload scenarios inducing significantly different operating conditions. In particular, we analyze the effect of variations in the incoming request rates $v$ and service rates $q$, which directly govern the system dynamics. In the first scenario ("flash-crowd/demand surge"), during the time interval $t \in [20, 38]$, the incoming request rates are sharply increased $(2-3\times$ compared to the rest of the time windows) headed toward higher-QoS and higher-security SFCs, i.e., $(2,3)$, $(3,2)$ and $(3,3)$. This models situations such as emergency events or sudden concentration of premium service usage, where demand becomes both bursty and skewed toward critical services. In the second scenario ("service degradation"), during the interval $t \in [15, 32]$, the service rates of selected medium- and high-QoS SFCs, i.e., $(2,2)$, $(2,3)$ and $(3,2)$, are significantly reduced $(0.35-0.5\times$ compared to the rest of the time windows), emulating VNF degradation or partial failures. This is followed by a gradual recovery phase for $t \in [33, 42]$ $(0.7-0.75\times$ compared to the rest of the time windows). In parallel, demand for higher-priority services is moderately increased $(1.4-1.5\times$ compared to the rest of the time windows), creating a capacity-constrained operating regime.




(a) QoS Violations in the flash-crowd/demand surge scenario.

(b) System Cost in the flash-crowd/demand surge scenario.

(c) QoS Violations in the service degradation scenario.

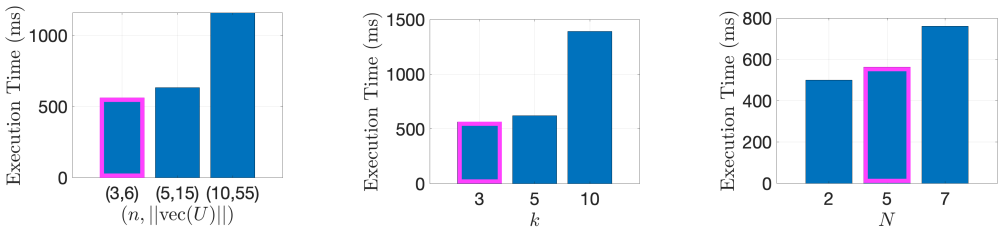
(d) System Cost in the service degradation scenario.

Figure 10: Comparative performance under additional stress-test workload scenarios: (a)-(b) flash-crowd/demand surge, (c)-(d) service degradation.

The results are presented in Figure 10. We observe that in the flash-crowd scenario, all baseline methods exhibit increased violation rates (Figure 10a), with EDF reaching approximately 21% and QSSFC around 18%, while PSG performs somewhat better at approximately 15%. The proposed MCIS-MPC does not incur any violations. In terms of cost (Figure 10b), EDF incurs the highest penalty (slightly above 100), reflecting its inability to effectively redistribute load under bursty conditions. PSG reduces the cost to approximately 45, while QSSFC and MCIS-MPC achieve the lowest values (around 32), indicating more efficient scheduling decisions. The proposed solution marginally outperforms the QSSFC in cost reduction as well. In the service degradation scenario, the differences become even more pronounced. In Figure 10c we observe that the violation rates of the baselines increase further (EDF $\approx 28\%$, PSG $\approx 27\%$, QSSFC $\approx 24\%$), while MCIS-MPC again incurs no violations. This corresponds to an order-of-magnitude improvement in QoS compliance. At the same time, MCIS-MPC preserves a low cost comparable to but still lower than QSSFC ($\approx 32$), as depicted in Figure 10d while EDF again exhibits significantly higher cost ($\approx 100$). PSG achieves a relatively higher cost as well ($\approx 49$). This demonstrates that MCIS-MPC is able to reallocate traffic effectively under reduced service capacity, rather than merely reacting to demand increases.

Overall, the results demonstrate that the proposed framework remains robust under substantial perturbations of both traffic-demand parameters and service-rate parameters, maintaining low QoS violations and competitive operational cost across significantly different operating regimes.

### 6.5. *Runtime complexity and scalability analysis*




(a) Effect of $n$,$||\,\mathrm{vec}(U)||$.

(b) Effect of $k$.

(c) Effect of $N$.

Figure 11: Runtime execution, for different state dimensions $n$, control inputs per subsystem $||\,\mathrm{vec}(U)||$, numbers of subsystems $k$ and control horizons $N$. The baseline configuration is in violet.

To further investigate the scalability of the proposed framework, we analyze the execution time of the online MPC optimization under varying problem dimensions. In particular, we study the effect of the number of QoS levels (state dimensions) $n$, number of control inputs per subsystem $||\,\mathrm{vec}(U)||$, number of subsystems $k$, and prediction horizon depth $N$ on the runtime performance, as the optimization complexity increases with them. We emphasize that the contribution of our work is not large-scale optimization over hundreds

or thousands of states, but rather the investigation of formally assured MPC-based runtime management for moderate-scale SFC configurations. We also note that the repeated application of the max operator in Eq. (23) across the prediction horizon results in a mixed-integer linear programming (MILP) formulation. While MILPs are NP-hard in the worst case, modern solvers such as `Gurobi` exploit branch-and-bound techniques, linear relaxations, presolve reductions, and heuristics to achieve tractable runtimes for moderate-scale instances such as those considered in this work.

Figure 11 presents the runtime execution analysis under different problem dimensions. The baseline configuration used throughout this section so far, highlighted in violet, corresponds to three QoS levels, six control inputs per subsystem, three security subsystems, and prediction horizon $N = 5$, resulting in an average execution time of approximately $560ms$. Figure 11a illustrates the effect of increasing the number of QoS levels and corresponding control inputs while keeping the remaining parameters fixed. Increasing the configuration from $(3, 6)$ to $(5, 15)$ increases the execution time from approximately $560ms$ to $640ms$. Increasing the dimensionality further to $(10, 55)$ results in execution times close to $1.15s$. This increase is expected since additional QoS levels enlarge the scheduling space and increase the number of optimization variables. Similarly, Figure 11b analyzes the effect of increasing the number of subsystems while keeping the remaining parameters fixed. Increasing the number of subsystems from $k = 3$ to $k = 5$ results in a moderate increase in runtime from approximately $560ms$ to $630ms$. Increasing the number of subsystems further to $k = 10$ increases the execution time to approximately $1.4s$, reflecting the additional coupling and optimization complexity introduced by larger multi-security configurations. Finally, Figure 11c shows the effect of varying the prediction horizon depth. Reducing the horizon from $N = 5$ to $N = 2$ decreases the average execution time from approximately $560ms$ to $500ms$, while increasing the horizon to $N = 7$ increases the runtime to approximately $760ms$. This behavior reflects the expected tradeoff between optimization depth and computational overhead. Overall, the main scalability limitations arise from the set-based computations and mixed-integer optimization structure inherent to formally assured control methods, however, the results indicate that the proposed framework remains computationally tractable for practically relevant multi-level QoS/security SFC deployments.

## 7. Conclusion

### *7.1. Contribution summary*

Aligned with existing literature providing a foundation for understanding the realization of Network Assurance (NA), our work deals with performance guarantees while catering for the flexibility, security, efficiency, and scalability of networking infrastructure. Our framework provides strict NA in the context of request scheduling, tailored specifically for multi-tenant SFC environments. We tackle two crucial challenges, namely, i) the scalability of the scheduling mechanism, by decomposing the system into subsystems, each managed by hierarchically connected MPCs instead of a single centralized controller and ii) the *a priori* guarantee of QoS satisfaction; given that in the examined problem exogenous signals are coupled with the control inputs, we calculate the maximal controlled invariant sets (MCIS) for each subsystem. Utilising these sets as constraints in the associated MPC problem we enforce meeting the performance specifications of the networking infrastructure. This is the first work, to our knowledge, that guarantees multiple network service requirements for the lifetime of a request, under the realistic assumption of externally induced uncertainties.

### *7.2. Limitations and future work*

Common to set-based methods, a remaining challenge lies in the efficient computation of controlled invariant sets for higher-order systems. While the proposed framework remains computationally tractable for moderate-scale multi-level QoS/security SFC deployments, the complexity of invariant-set computations and mixed-integer optimization grows significantly with the dimensionality of the scheduling problem. As future work, we intend to investigate relaxed notions of controlled invariance that enable more efficient algorithmic constructions, e.g., [47, 48, 49], as well as decomposition-based and approximate set representations. Another limitation of the current formulation is the assumption of accurate knowledge of the service and incoming request rates. Extending the framework to explicitly account for estimation uncertainty and model mismatch

is part of our ongoing research, following robust and tube-based MPC approaches, e.g., [50, 51, 52]. Finally, the current framework assumes runtime management within a single administrative domain with global visibility over the SFC infrastructure. Extending the proposed approach to multi-domain environments constitutes an important direction for future work. Such settings introduce additional challenges related to distributed coordination, partial observability, heterogeneous domain policies, and limited information sharing across administrative boundaries.


### Acknowledgement

This work was supported by the NSERC ALLRP 580715-22 program, and by the European Union's Horizon Europe research and innovation programme under the project UniMaaS with grant agreement No 101177842.


### CRediT authorship contribution statement

Marios Avgeris: Conceptualization, Methodology, Software, Writing – original draft, Validation, Investigation. Nikolaos Athanasopoulos: Conceptualization, Methodology, Formal Analysis, Writing – original draft. Aris Leivadeas: Writing – review & editing, Investigation, Supervision, Project administration, Funding acquisition. Ioannis Lambadaris: Writing – review & editing, Project administration, Funding acquisition.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

No data was used for the research described in the article.

### References


[1] D. Zeng, N. Ansari, M.-J. Montpetit, E. M. Schooler, D. Tarchi, Guest editorial: In-network computing: Emerging trends for the edge-cloud continuum, IEEE Network 35 (5) (2021) 12–13. doi:10.1109/MNET.2021.9606835.

[2] X. Zheng, 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. doi:10.23919/CNSM52442.2021.9615580.

[3] G. Li, et al., Efficient deployment and scheduling of shared vnf instances in mobile edge computing networks, IEEE Internet of Things Journal (2024).

[4] W. Attaoui, E. Sabir, H. Elbiaze, M. Guizani, Vnf and cnf placement in 5g: Recent advances and future trends, IEEE Transactions on Network and Service Management 20 (4) (2023) 4698–4733.

[5] M. Di Mauro, M. Longo, F. Postiglione, Availability evaluation of multi-tenant service function chaining infrastructures by multidimensional universal generating function, IEEE Transactions on Services Computing 14 (5) (2018) 1320–1332.

[6] M. Verma, D. Behl, P. Jayachandran, A. Singh, M. Thomas, A reliability assurance framework for cloud-native telco workloads, in: 2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS), 2023, pp. 201–203. doi:10.1109/COMSNETS56262.2023.10041285.

[7] E. B. Fernandez, A. Brazhuk, A critical analysis of zero trust architecture (zta), Computer Standards & Interfaces 89 (2024) 103832. doi:https://doi.org/10.1016/j.csi.2024.103832.
URL https://www.sciencedirect.com/science/article/pii/S0920548924000011

[8] J. Wang, J. Liu, J. Li, N. Kato, Artificial intelligence-assisted network slicing: Network assurance and service provisioning in 6g, IEEE Vehicular Technology Magazine 18 (1) (2023) 49–58. doi:10.1109/MVT.2022.3228399.

[9] M. Avgeris, A. Leivadeas, N. Athanasopoulos, I. Lambadaris, M. Falkner, Model predictive control for automated network assurance in intent-based networking enabled service function chains, in: NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2023, pp. 1–7.

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

[11] N.-N. Dao, N. H. Tu, T.-D. Hoang, T.-H. Nguyen, L. V. Nguyen, K. Lee, L. Park, W. Na, S. Cho, A review on new technologies in 3gpp standards for 5g access and beyond, Computer Networks 245 (2024) 110370.

[12] E. Coronado, et al., Zero touch management: A survey of network automation solutions for 5g and 6g networks, IEEE Comm. Surveys & Tutorials 24 (4) (2022) 2535–2578.

[13] R. Asensio-Garriga, A. Molina Zarca, J. Ortiz, A. Hermosilla, H. R. Pascual, A. Pastor, A. Skarmeta, Zsm framework for autonomous security service level agreement life-cycle management in b5g networks, Future Internet 17 (2) (2025). doi:10.3390/fi17020086.
URL https://www.mdpi.com/1999-5903/17/2/86

[14] 3GPP, 3rd generation partnership project; technical specification group services and system aspects; management and orchestration; management services for communication service assurance; requirements (release 18), Tech. Rep. TS 28.535 V18.0.0 (2024-04) (2024).

[15] D. Soldani, et al., ebpf: A new approach to cloud-native observability, networking and security for current (5g) and future mobile networks (6g and beyond), IEEE Access 11 (2023) 57174–57202.

[16] S. K. Perepu, J. P. Martins, R. Souza, K. Dey, Intent-based multi-agent reinforcement learning for service assurance in cellular networks, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 2879–2884.

[17] T. A. Khan, K. Abbas, M. Afaq, W.-C. Song, Accelerating zero-touch automation and optimization of beyond 5g services: deep learning and intent-based networking fusion: Ta khan et al., The Journal of Supercomputing 81 (7) (2025) 833.

[18] K. Abbas, Y. Cho, M. Afaq, A. Nauman, J.-H. Yoo, J. W.-K. Hong, W.-C. Song, Ibn-ztsa: Ai-ibn for zero touch service automation of b5g terrestrial and non-terrestrial networks, IEEE Communications Standards Magazine (2025).

[19] M. Xie, et al., Service assurance architecture in nfv, in: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE, 2017, pp. 229–235.

[20] E. Battiston, D. Moro, G. Verticale, A. Capone, Chima: A framework for network services deployment and performance assurance, in: 2022 IEEE 8th International Conference on Network Softwarization (NetSoft), IEEE, 2022, pp. 163–170.

[21] A. Nouruzi, A. Zakeri, M. R. Javan, N. Mokari, R. Hussain, S. A. Kazmi, Online service provisioning in nfv-enabled networks using deep reinforcement learning, IEEE Trans. on Network and Service Management 19 (3) (2022) 3276–3289.

[22] T.-H. Lei, Y.-T. Hsu, I.-C. Wang, C. H.-P. Wen, Deploying qos-assured service function chains with stochastic prediction models on vnf latency, in: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE, 2017, pp. 1–6.

[23] W. You, X. Xie, H. Wang, J. Xia, V. Stojanovic, Relaxed model predictive control of ts fuzzy systems via a new switching-type homogeneous polynomial technique, IEEE Transactions on Fuzzy Systems 32 (8) (2024) 4583–4594.

[24] L. Gao, Z. Zhuang, H. Tao, Y. Chen, V. Stojanovic, Non-lifted norm optimal iterative learning control for networked dynamical systems: A computationally efficient approach, Journal of the Franklin Institute 361 (15) (2024) 107112.

[25] M. Avgeris, D. Dechouniotis, N. Athanasopoulos, S. Papavassiliou, Adaptive resource allocation for computation offloading: A control-theoretic approach, ACM Transactions on Internet Technology (TOIT) 19 (2) (2019) 1–20.

[26] J. Halpern, C. Pignataro, Service function chaining (sfc) architecture, Tech. rep. (2015).

[27] E. Vlahakis, R. Jungers, N. Athanasopoulos, S. McLoone, Aimd-inspired switching control of computing networks, IEEE Transactions on Control of Network Systems (2023).

[28] J. Leroux, S. Schmitz, Demystifying reachability in vector addition systems, in: 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science, 2015, pp. 56–67. doi:10.1109/LICS.2015.16.

[29] D. Dechouniotis, N. Leontiou, N. Athanasopoulos, A. Christakidis, S. Denazis, A control-theoretic approach towards joint admission control and resource allocation of cloud computing services, International Journal of Network Management 25 (3) (2015) 159–180.

[30] V. D. Blondel, J. N. Tsitsiklis, A survey of computational complexity results in systems and control, Automatica 36 (9) (2000) 1249–1274.

[31] R. J. Hanson, Linear least squares with bounds and linear constraints, SIAM Journal on scientific and statistical computing 7 (3) (1986) 826–834.

[32] W. Ben-Ameur, H. Kerivin, Routing of uncertain traffic demands, Optimization and Engineering 6 (2005) 283–313.

[33] F. Blanchini, S. Miani, et al., Set-theoretic methods in control, Vol. 78, Springer, 2008.

[34] C. Belta, B. Yordanov, E. A. Gol, Formal methods for discrete-time dynamical systems, Vol. 89, Springer, 2017.

[35] S. V. Rakovic, E. C. Kerrigan, D. Q. Mayne, J. Lygeros, Reachability analysis of discrete-time systems with disturbances, IEEE Transactions on Automatic Control 51 (4) (2006) 546–561.

[36] G. M. Ziegler, Lectures on polytopes, Vol. 152, Springer Science & Business Media, 2012.

[37] R. T. Rockafellar, Convex analysis, Vol. 28, Princeton university press, 1997.

[38] J. Lv, X. Li, T. Li, Network traffic prediction and applications based on time series model, in: International Conference on Intelligent Computing, Springer, 2007, pp. 1306–1315.

[39] M. Choi, H.-G. Kim, J.-H. Yoo, J. W.-K. Hong, Network traffic prediction and auto-scaling of sfc using temporal fusion transformer, in: Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE, 2023, pp. 131–136.

[40] P. Barford, D. Plonka, Characteristics of network traffic flow anomalies, in: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, 2001, pp. 69–73.

[41] X. Ge, F. Yang, Q.-L. Han, Distributed networked control systems: A brief overview, Information Sciences 380 (2017) 117–131. doi:https://doi.org/10.1016/j.ins.2015.07.047.

[42] A. Leivadeas, 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. doi:10.1109/ICIN51074.2021.9385553.

[43] A. Leivadeas, M. Falkner, 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), IEEE, 2020, pp. 1–6.

[44] J. A. Stankovic, M. Spuri, K. Ramamritham, G. C. Buttazzo, Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms, Vol. 460, Springer Science & Business Media, 2012.

[45] S. Azizi, M. Shojafar, J. Abawajy, R. Buyya, Deadline-aware and energy-efficient iot task scheduling in fog computing systems: A semi-greedy approach, Journal of network and computer applications 201 (2022) 103333.

[46] G. Li, H. Zhou, B. Feng, G. Li, S. Yu, Automatic selection of security service function chaining using reinforcement learning, in: 2018 IEEE Globecom Workshops (GC Wkshps), IEEE, 2018, pp. 1–6.

[47] N. Athanasopoulos, A. I. Doban, M. Lazar, On constrained stabilization of discrete-time linear systems, in: 21st Mediterranean conference on control and automation, IEEE, 2013, pp. 830–839.

[48] N. Athanasopoulos, M. Lazar, Scalable stabilization of large scale discrete-time linear systems via the 1-norm, IFAC Proceedings Volumes 46 (27) (2013) 277–284.

[49] S. Olaru, M. Soyer, Z. Zhao, C. E. Dórea, E. Kofman, A. Girard, From relaxed constraint satisfaction to p-invariance of sets, IEEE Transactions on Automatic Control 69 (10) (2024) 7036–7042.

[50] D. Q. Mayne, M. M. Seron, S. V. Raković, Robust model predictive control of constrained linear systems with bounded disturbances, Automatica 41 (2) (2005) 219–224.

[51] D. Q. Mayne, E. C. Kerrigan, E. Van Wyk, P. Falugi, Tube-based robust nonlinear model predictive control, International journal of robust and nonlinear control 21 (11) (2011) 1341–1353.

[52] D. Limón, I. Alvarado, T. Alamo, E. F. Camacho, Robust tube-based mpc for tracking of constrained linear systems with additive disturbances, Journal of Process Control 20 (3) (2010) 248–260.

**Appendix A. Constraint set $\mathcal{W}$ and subsystems data** (2)**, Section 6.**

$$G_v^{'}=\begin{bmatrix}0&0&0\\1&1&1\\0&0&2\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}, G_q^{'}=\begin{bmatrix}-1&-1&-1\\0&0&0\\0&0&-1\\0&0&0\\0&0&0\\0&0&0\end{bmatrix}, w'=\begin{bmatrix}-25\\27\\0\\15\\7\\5\end{bmatrix},$$

$$G_u^{'}=\begin{bmatrix}1&1&1&0&0&0\\-1&-1&-1&0&0&0\\0&0&0&1&1&0\\0&0&0&-1&-1&0\\0&0&0&0&0&1\\0&0&0&0&0&-1\end{bmatrix}, G_v^{''}=\begin{bmatrix}-1&0&0\\1&0&0\\0&1&0\\0&-1&0\\0&0&1\\0&0&-1\end{bmatrix},$$

$$G_u=\begin{bmatrix}0^{6\times 6}\\G_u'\end{bmatrix}, G_v=\begin{bmatrix}G_v^{'}\\G_v^{''}\end{bmatrix}, G_q=\begin{bmatrix}G_q^{'}\\0^{6\times 3}\end{bmatrix}, w=\begin{bmatrix}w'\\0^{6\times 1}\end{bmatrix}.$$

The linear inequality constraints induced by matrix $G_v^{'}$ correspond to incoming flows patterns. The constraints on $G_u^{'}$, $G_v^{''}$ correspond to the equalities that define the scheduling architecture. The subsystem parameters are identified as $a^1=a^2=a^3=1$,

$$B^1=\alpha\begin{bmatrix}0.10&0&0\\0.07&0.05&0\\0.39&0.09&0.24\end{bmatrix}, e^1=\begin{bmatrix}0.10\\0.05\\0.24\end{bmatrix},$$

$$B^2=\alpha\begin{bmatrix}0.10&0&0\\0.06&0.05&0\\0.22&0.63&0.29\end{bmatrix}, e^2=\begin{bmatrix}0.10\\0.05\\0.29\end{bmatrix},$$

$$B^3=\alpha\begin{bmatrix}0.10&0&0\\0.05&0.05&0\\0.10&0.64&1.45\end{bmatrix}, e^3=\begin{bmatrix}0.10\\0.05\\1.45\end{bmatrix}.$$