Modeling the Cost of Flexibility in Communication Networks

Alberto Martínez Alba*, Péter Babarczi*†, Andreas Blenk*, Mu He*, Patrick Kalmbach*, Johannes Zerwas*, and Wolfgang Kellerer*

*Chair of Communication Networks, Technical University of Munich, Germany

E-mail: {alberto.martinez-alba, andreas.blenk, mu.he, patrick.kalmbach, johannes.zerwas, wolfgang.kellerer}@tum.de

†MTA-BME Future Internet Research Group, Dept. of Telecommunications and Media Informatics, Budapest University of Technology and Economics (BME), Hungary, E-mail: babarczi@tmit.bme.hu

***Abstract*—Communication networks are evolving towards a more adaptive and reconfigurable nature due to the evergrowing demands they face. A framework for measuring network flexibility has been proposed recently, but the cost of rendering communication networks more flexible has not yet been mathematically modeled. As new technologies such as software-defined networking (SDN), network function virtualization (NFV), or network virtualization (NV) emerge to provide network flexibility, a way to estimate and compare the cost of different implementation options is needed. In this paper, we present a comprehensive model of the cost of a flexible network that takes into account its transient and stationary phases. This allows network researchers and operators to not only qualitatively argue about their new flexible network solutions, but also to analyze their cost for the first time in a quantitative way.**

***Index Terms*—cost, flexibility, communication networks, adaptation**

I. Introduction

In recent years, increasing the adaptability and scalability of current communication networks has attracted considerable research effort. This is due to the evergrowing number of connected devices and the emergence of new use cases and applications in all types of networks. For instance, with the emergence of Internet of Things (IoT), the number of deployed autonomous devices is estimated to grow from 12 billion in 2015 to hundreds of billions by 2025 [1]. These devices will cover a large number of IoT use cases, such as industrial automation, self-driving cars, and smart-city sensors [2]. Regarding mobile networks, 5G and 6G technologies aim at accomplishing high resource efficiency while being able to quickly adapt to changes in user mobility and traffic patterns [3], [4]. In addition, a major concern for data center network operators is to increase the agility and programmability of such networks to face new applications [5].

A popular option among researchers and operators to increase network adaptability is network softwarization, as in software-defined networking (SDN), network function virtualization (NFV), and network virtualization (NV) [6]. Network softwarization entails replacing hardware with software entities, which helps to prevent the stiffness of hardware-based solutions, whose reconfiguration is slow and costly. Instead, software can be reconfigured and scaled with ease, paving the way for larger and more flexible communication networks.

Being relatively new, the concept of network softwarization is still a greenfield in network research. The novel ability of softwarized networks to quickly respond to demand changes even spurs dedicated research to measure this ability. Indeed, recent works even propose new metrics to quantify the responsiveness of a network under changing demands: the so-called *network flexibility* [7]. Regardless of other performance indicators, the purpose of this new metric is to indicate how quickly and costly a softwarized network is able to adapt to a changing demand. This intends to provide a common ground to compare the effectiveness of alternative solutions whose performance may be similar in static conditions.

Network flexibility is a desirable characteristic that network operators want to maximize. Indeed, the high flexibility of a network not only measures its good responsiveness in case of demand changes, but it can also be used as advertisement to attract potential users. In addition, it also serves as an indicator for the ability of the network to withstand changes in future demand trends. However, increasing network flexibility does not come for free. Flexibility is affected by a trade-off that also applies to many other performance metrics: the higher its value, the higher the costs, yet the higher the revenue. The reason why this happens is simple to understand. On the one hand, in order to increase the flexibility, the operator may need to invest in better equipment and consume more resources, which can be costly. On the other hand, a high network flexibility is linked with increased revenues: more supported users, less sensitivity to failed links, etc. The combination of both opposite trends lays out the classical design problems: maximizing performance while keeping the costs, minimizing the costs while keeping the performance, or find the best performance/cost combination of the Pareto frontier.

The conventional approach to model the cost of a communication network, as well as any other engineering system, is to divide its total cost of ownership (TCO) into capital

This work is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 647158 - FlexNets "Quantifying Flexibility for Communication Networks"). The work of P. Babarczi was supported by Project no. 134604 that has been implemented with the support provided by the National Research, Development and Innovation Fund of Hungary, financed under the FK_20 funding scheme.

expanses (CAPEX) and operating expenses (OPEX) [8]. For networks facing slow-varying environments, the OPEX can be calculated from its expected operating state, i. e., from the average revenue and rate of resource consumption [9], [10]. Nevertheless, describing the OPEX of a network that faces a constantly changing environment is substantially more challenging. On the one hand, it may be difficult to estimate the expected operating state, as it is the result of a succession of changing demands. On the other hand, the adaptation itself is also costly, and thus is has to be included in the OPEX.

Given this scenario, we envision a more powerful cost model whose components directly characterize network adaptations. This model can be used to accurately predict the total cost of a network whose adaptations are in the same order of magnitude as the changes in the environment, thus showing that adaptation may be worthwhile even in "race conditions". Previous work [11], [12] tackles network dynamicity, but, to the best of our knowledge, no cost model is available to assess profitability in these conditions. In addition, this cost model can be used to find out the optimal adaptation strategy, identify limiting factors, and test for profitability in future scenarios.

In this paper, we use a probability theory framework to derive a cost model of a flexible network. Since it captures the internal trade-offs that affect flexibility and cost, it offers a much richer point of view with respect to the conventional analysis of the TCO. Moreover, the model can be used to make predictions or take decisions that affect network design, such as selecting the least costly deployment option. Although we tailor this model for communication networks, the model is provided with mathematical rigor so that it can be extended to other flexible systems that accept similar definitions.

The rest of the paper is as follows. In Sec. II we introduce the system model, including a formal definition of network flexibility. In Sec. III we present the complete cost model. Sec. IV contains examples of how the model can be applied to real networks. Finally, Sec. V concludes the paper.

II. System Model

In this section, we introduce the concepts required to derive the cost model of a flexible network.

A. Network states and demands

We consider a scenario consisting of a softwarized, configurable communication network managed by a *network operator* to achieve a profitable purpose, such as providing connectivity to users, carrying information within a data center, or managing virtual network slices. The instantaneous configuration of the network is referred to as the *state* $s \in \mathbb{S}$ of the network, where $\mathbb{S}$ is the set of all possible states that can be achieved. For example, the routing tables in the network switches, the location of virtual functions, or the physical resources allocated to a network slice can be used as the state.

The conditions on which the network operates are modeled by the demand $d \in \mathbb{D}$, where $\mathbb{D}$ is the set of all possible demands. The demand includes all parameters that affect the network's profitability but cannot be modified by the network. These parameters can describe the external environment (such as the number of connected users or the requested virtual flows), but they also include any internal configuration that may change out of the network's control (such as the topology graph of active nodes and links in a resilient network).

We say that a demand is *satisfied* if the network state is able to fulfill the expectations that this demand generates. For instance, a demand consisting of a flow request between two network points is satisfied if the intermediate nodes can forward the packets correctly between these points. Those states satisfying a given demand are called *valid* states, whereas any other state is an *invalid* state. We define the function $\mathbb{V}(d)$ to relate demand d to its set of valid states:

$$\mathbb{V}(d) : \mathbb{D} \mapsto \wp(\mathbb{S}), \tag{1}$$

where $\wp(\mathbb{S})$ is the power set of $\mathbb{S}$. If $\mathbb{V}(d) = \emptyset$, we say that the demand d is *unsatisfiable*.

In a flexible network, demands and states are subject to change over time. From the definition of demand, it follows that the network cannot accurately predict nor prevent demand changes. We model a sequence of demands within time interval $(0, \tau)$ as a discrete stochastic process $\{\mathcal{D}_i\}_{i \in \mathbb{Z}}$ on the sample space $\mathbb{D}$, where i is an arbitrary time-ordered integer index. We also define the sequence of states $\{\mathcal{S}_i\}_{i \in \mathbb{Z}}$ on the sample space $\mathbb{S}$. We model the duration of each demand $\mathcal{D}_i = d_i$ by the random variable $\mathcal{T}_i$ on the sample space $\mathbb{R}^+$, and hence we define the stochastic process $\{\mathcal{T}_i\}_{i \in \mathbb{Z}}$ as the sequence of durations of each demand. Assuming that $\{\mathcal{T}_i\}$ is stationary, it can be described by its marginal cumulative distribution function (CDF) $F_{\mathcal{T}}(t)$. Processes $\{\mathcal{D}_i\}$ and $\{\mathcal{T}_i\}$ fully describe the demands over time, as every observed demand $\mathcal{D}_i = d_i$ is associated with a duration $\mathcal{T}_i = t_i$.

A change in the network state is the result of a conscious network decision, which is taken to address a demand change. Hence, the sequence of states is determined by the sequence of demands. For convenience, we introduce the following notation to represent a demand and a state change, respectively:

$$\widetilde{d_i} \triangleq \langle d_i, d_{i+1} \rangle , \quad \widetilde{s_i} \triangleq \langle s_i, s_{i+1} \rangle . \tag{2}$$

We consider that an *adaptation* consists of a demand change $\widetilde{d_i}$ and its corresponding state change $\widetilde{s_i}$. From a modeling point of view, we associate every demand change with a state change, although in practice it may happen that $s_i = s_{i+1}$ if there is no effective state change.

B. The adaptation process

After a noticing a demand change $\widetilde{d_i}$, a flexible network needs to perform two tasks. First, it needs to run an *adaptation algorithm* to find the most appropriate state s_{i+1} to satisfy the new demand d_{i+1}. Formally, we model the outcome of this algorithm by means of the *adaptation function*:

$$a(d) : \mathbb{D} \mapsto \mathbb{S}, \tag{3}$$

so that $s_{i+1} = a(d_{i+1})$. Since finding this new state may be computationally hard, we need to account for the time and cost required to do this, as they may impact the overall profitability.

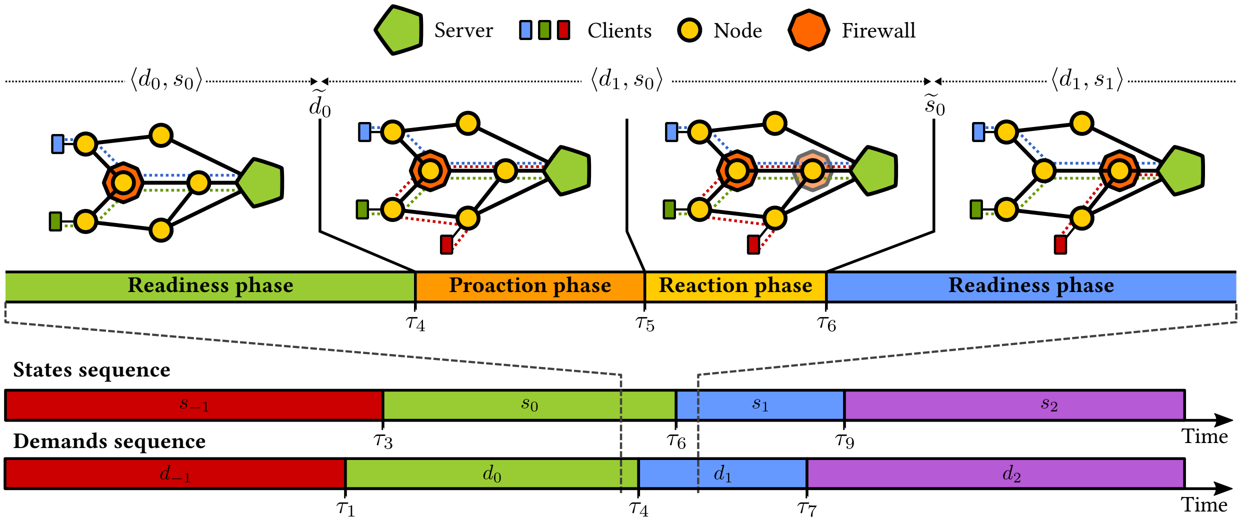

Fig. 1: Adaptation phases traversed by a flexible network. The network operator is in charge of providing connectivity between server and clients while ensuring that all packets go through the firewall, whose location can be dynamically chosen.

We refer to the former as *proaction time* z_i^P and to the latter as *proaction cost* c_i^P. In addition, once the network has found the new state, it has to move from the old state to the new one. We refer to the time and cost required to change the state as *reaction time* z_i^R and *reaction cost* c_i^R, respectively.

Overall, the time difference between a demand change $\widetilde{d}_i$ and its corresponding state change $\widetilde{s}_i$ is the *action time* $z_i = z_i^P + z_i^R$. As with the sequences of states and demands, we define the discrete stochastic process $\{\mathcal{Z}_i\}_{i\in\mathbb{Z}}$ on the sample space $\mathbb{R}^+$ to model the sequence of action times $\mathcal{Z}_i = z_i$ for every time index i. Assuming stationarity, this process can be characterized by its marginal CDF $F_{\mathcal{Z}}(z)$, i. e., the distribution of the durations of each adaptation. We define in the same manner processes $\{\mathcal{Z}_i^P\}_{i\in\mathbb{Z}}$ and $\{\mathcal{Z}_i^R\}_{i\in\mathbb{Z}}$ for the proaction and reaction times, respectively. Similarly, we denote the *action cost* as $c_i = c_i^P + c_i^R$, which reflects the total effort of addressing demand change $\widetilde{d}_i$ and realizing state change $\widetilde{s}_i$, and define the discrete stochastic process $\{\mathcal{C}_i\}_{i\in\mathbb{Z}}$ on the sample space $\mathbb{R}^+$ to model action times $\mathcal{C}_i = c_i$, whose marginal CDF is $F_{\mathcal{C}}(c)$.

If at any time instant τ the network is satisfying the current demand, we say that the network is in the *readiness phase*. The cost per time unit associated with operating the system at this phase is referred to as the *readiness cost* k_j for an arbitrary time index j, which can be expressed as a function of the active demand and state. This is explained in detail in Sec. III-B. The readiness cost is affected by the amount of resources consumed in the current state and the revenue obtained from demand satisfaction. As a result, not being able to satisfy a demand mainly affects this cost component.

In order to clarify the meaning of the aforementioned definitions, we present an exemplary adaptation timeline in Fig. 1. This figure shows the observed demands and the states implemented by a network providing connectivity between a server and a set of clients. This connectivity is provided while enforcing a security policy: any packet between the server and the clients must go through a virtual firewall, whose location can be changed during runtime. The optimal firewall location is the one that minimizes the number of links traversed by all packets, thus minimizing latency. Between τ_3 and τ_4, the network is operating in the readiness phase: demand d_0 (clients blue and green) is being satisfied by state s_0 with minimal latency. The revenue obtained from satisfying this demand and the cost of using network resources (links, nodes, CPU, etc.) is reflected by the readiness cost. At time τ_4, the demand changes to d_1: the red client connects to the network. After noticing the demand change, the network realizes that firewall location may not be optimal anymore. As a result, it triggers the adaptation algorithm to find out the optimal state for demand d_1. The time during which the adaptation algorithm is running is the *proaction phase*, and the additional cost associated to it (due to higher resource consumption) is the *proaction cost*. At time τ_5, the adaptation algorithm has converged and returned a state $s_1 \neq s_0$ featuring a new firewall location. Therefore, the network starts the procedure to migrate the firewall, hence starting the *reaction phase*, which lasts until the migration is completed at τ_6. Any additional cost associated to this phase is reflected by the *reaction cost*. The union of the proaction and reaction phases is the *action phase*, during which the active state is delayed with respect to the current demand.

C. Flexibility measure

The ability of a network to adapt to a changing environment has been tackled to some extent by previous literature. For instance, in [13], [14] a mathematical framework for a rigorous definition of network flexibility is provided. In particular, for a given demand sequence, network flexibility $\Phi(z, c)$ is defined as the ratio of satisfied demands within time limit z and cost limit c to the total number of demands. This definition can be easily connected with the present cost model, resulting in a more complete mathematical framework.

The intention of defining network flexibility is to measure the frequency of non-ideal responses to a demand change. Ideally, every demand change should result in a state change leading to a valid state. In real life, adaptation algorithms are not perfect and demands may be unsatisfiable, thus it could happen that the network cannot find a valid state for a new demand. As a result, we can split the sequence of demands $\{\mathcal{D}_i\}$ into two non-overlapping sequences of satisfied demands $\{\mathcal{D}_i^{\in}\}$ and unsatisfied demands $\{\mathcal{D}_i^{\notin}\}$ based on whether $a(d_i) \in \mathbb{V}(d_i)$ or not. From these sets, we can define the *maximum flexibility* φ of the network as

$$\varphi = \lim_{i\to\infty} \frac{|\{\mathcal{D}_i^{\in}\}|}{|\{\mathcal{D}_i\}|}, \tag{4}$$

where the operator $|\cdot|$ yields the total length of a sequence. The maximum flexibility φ is thus the ratio of satisfied demands to total demands, in the absence of the cost and time constraints.

III. Cost model of a flexible network

In this section, we analyze the components of the total cost in a flexible network and relate them to the flexibility framework defined in the previous section.

A. General definitions

As introduced in Sec. II-B, our cost model consists of three independent components: readiness, proaction, and reaction costs. These components can be straightforwardly combined to obtain the total cost of operating a network.

Definition III.1. *The total cost Q of operating a flexible network over a long time interval $(0, \tau)$ is*

$$Q = K + C^P + C^R, \tag{5}$$

where K, C^P, and C^R are the mean readiness cost, proaction cost, and reaction cost.

For notation convenience, these components reflect *cost over time* (in arbitrary monetary units per time unit), rather than absolute cost. Hence, the absolute cost of operating a network over interval $(0, \tau)$ is $Q\tau$.

In order to achieve a more powerful model, the total cost Q includes not only expenses, but also revenue coming from providing service to users. This revenue is modeled as negative cost, hence we say that the network is *profitable* over interval $(0, \tau)$ if and only if $Q < 0$. Although a network provider could charge users when they specifically request a service, nowadays a subscription-based revenue, in which users pay a flat rate for a service, is the dominant strategy [15], [16]. Thus, we model revenue as a part of the readiness cost, resulting in $C^P > 0$, $C^R > 0$ and $K < 0$ in a profitable network.

B. Readiness cost

The instantaneous readiness cost $k(s, d)$ is the cost of operating a network in state s under demand d. Formally:

$$k(s, d) : \mathbb{S} \times \mathbb{D} \mapsto \mathbb{R}^+. \tag{6}$$

In words, $k(s, d)$ reflects how well the network is satisfying demand d. It includes both cost and revenue of operating a state: resource consumption, user payment via subscriptions, penalizations for unsatisfied demands, etc. Thus, it is the only cost component that can take negative values, which implies that the network operator obtains a profit from operating in the current state. This fact leads to the definition of the *optimal adaptation function* $a^*(d_i)$, which returns the valid state that minimizes the readiness cost for demand d_i:

$$a^*(d_i) = \arg\min_s k(s, d_i), \tag{7a}$$

$$\text{s.t.} \quad s \in \mathbb{V}(d_i). \tag{7b}$$

In real scenarios, however, finding the optimal solution to this problem may be too time consuming. We thus consider a more general definition of the adaptation function $a(d_i)$, which approximates $a^*(d_i)$ but may return suboptimal states or may fail to find a valid state. The ability of the adaptation function to return a (possibly suboptimal) valid state is captured by the maximum flexibility φ as defined in (4) in Sec. II-C.

When the network adapts to a sequence of demands $\{\mathcal{D}_i\}$ via a sequence of states $\{\mathcal{S}_i\}$, this results in a sequence of readiness costs that can be modeled as the stochastic process $\{\mathcal{K}_j\}_{j\in\mathbb{Z}}$ for every different demand-state pair. Since $\{\mathcal{D}_i\}$ and $\{\mathcal{S}_i\}$ are stationary, it follows that $\{\mathcal{K}_j\}$ is also stationary. Therefore, the mean readiness cost can be defined as the expected value of this sequence:

$$K \triangleq \mathrm{E}\{\mathcal{K}\}. \tag{8}$$

Note that we use a different variable to index the elements of process $\{\mathcal{K}_j\}$ with respect to processes $\{\mathcal{D}_i\}$ and $\{\mathcal{S}_i\}$. This is due to the possible presence of multiple demand-state combinations that result in different readiness costs. To explain this, let us consider a system facing demand d_i by implementing state $s_i = a(d_i)$. The resulting readiness cost in this situation is $k_j = k(s_i, d_i)$ (with slight abuse of notation). When a new demand d_{i+1} is requested, the readiness cost changes to $k_{j+1} = k(s_i, d_{i+1})$, as state s_i may not satisfy the new demand, leading to degraded performance and higher cost. At this point, there are multiple possibilities for the next readiness cost value. It could happen that the network finds a valid state $s_{i+1} = a(d_{i+1})$ before the demand changes again, leading to $k_{j+2} = k(s_{i+1}, d_{i+1})$ after state change $\widetilde{s}_i$. Conversely, the system may be unable to find a valid state or a new demand may appear before the new state is implemented, leading to a new readiness cost value of $k_{j+2} = k(s_i, d_{i+2})$.

In order to model the cost resulting from the offset between demands and valid states, we define the *state delay* $x(\tau)$ at time instant τ as the index difference between current demand $d(\tau) = d_i$ and current state $s(\tau) = s_j$, such that $s_j = a(d_{i-x(\tau)})$. Similarly to $\{\mathcal{S}_i\}$ and $\{\mathcal{D}_i\}$, the sequence of state delays can be modeled by the discrete stochastic process $\{\mathcal{X}_j\}_{j\in\mathbb{Z}}$. The instantaneous state delay resulting from a sequence of demands and states is shown in Fig. 2. We denote the marginal pmf of $\{\mathcal{X}_j\}$ as $f_{\mathcal{X}}(x)$, which yields the overall probability of the network operating with state delay x.

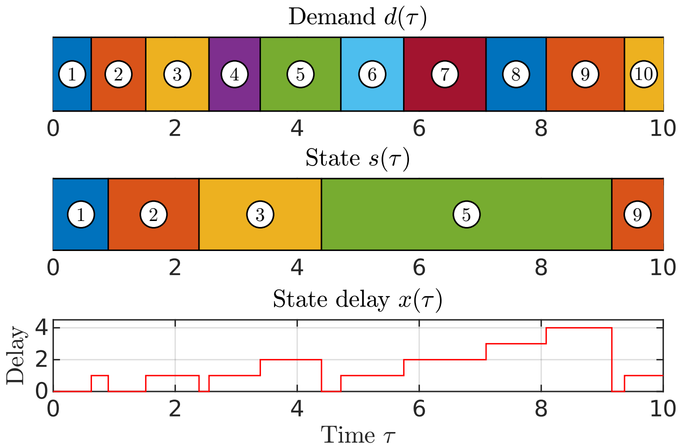

Fig. 2: Demands, states, and state delays experienced by an action-interrupting flexible network. The state delay is the instantaneous difference between demand and state indices (numbers in circles).

We define the *readiness degradation function* $K_{\mathcal{X}}(x) \triangleq \mathrm{E}\{\mathcal{K}|\mathcal{X}=x\}$ as the mean readiness cost when the state delay is $\mathcal{X} = x$. This function characterizes the performance of the network when dealing with delayed states. An example of such a function is shown in Sec. IV-B. In a properly designed network, the mean readiness cost is lowest when state delay is $\mathcal{X} = 0$, that is, when the network implements a valid state. Moreover, the cost of a state should monotonically grow with the state delay, reflecting that the demand becomes, on average, increasingly different from the last satisfied demand. Formally, this implies that $K_{\mathcal{X}}(x_2) \geq K_{\mathcal{X}}(x_1)$ if and only if $x_2 \geq x_1$. This leads to the following conclusion.

Lemma III.2. *A necessary condition for a network to be profitable is* $K_{\mathcal{X}}(0) < 0$.

The proof for Lemma III.2 is trivial, as it implies that a profitable network ($Q < 0$) requires at least that operating in valid states is profitable. Knowing this fact, the following lemma provides a method to compute the mean readiness cost.

Lemma III.3. *The mean readiness cost K of a flexible network can be calculated in terms of $K_{\mathcal{X}}(x)$ as:*

$$K = \sum_{x=0}^{\infty} K_{\mathcal{X}}(x) f_{\mathcal{X}}(x). \tag{9}$$

Proof. Eq. (9) follows directly from the application of the law of total expectation [17]. □

As mentioned before, $K_{\mathcal{X}}(x)$ is a characteristic function of the analyzed network and has to be measured, simulated, or theoretically derived for each case. The pmf $f_{\mathcal{X}}(x)$ can be obtained from $F_{\mathcal{T}}(t)$ and $F_{\mathcal{Z}}(z)$, that is, from the distributions of demand duration and action time. Nonetheless, the resulting expression for $f_{\mathcal{X}}(x)$ is affected by the behavior of the network when a new demand change appears during the action phase, that is, while looking for or moving to a new state.

We refer to a network as *action-persistent* if its action phases are not interrupted by a change in the demand. That is, an action-persistent network carries on with the action phase and realizes the state change even if the new state is already delayed from the start. Conversely, an *action-interrupting* network stops and resets its action phase if a new demand appears therein. As a consequence, an action-interrupting network only realizes state changes leading to valid, non-delayed states. Fig. 2 is an example of an action-interrupting network behavior. Owing to space limitations, in this work we only show the derivation of $f_{\mathcal{X}}(x)$ for action-interrupting networks, leaving the analysis of action-persistent networks for future work. We also argue that action-interrupting networks are the best option for operators who prefer to implement valid states rather than not to interrupt adaptations.

In our path to calculate $f_{\mathcal{X}}(x)$, we define the random variable $\mathcal{R}$ as the time difference between any instant in the considered interval $(0, \tau)$ and the most recent demand change. The probability density function (pdf) of this variable is provided in the following lemma.

Lemma III.4. *The pdf $f_{\mathcal{R}}(r)$ of $\mathcal{R}$ is*

$$f_{\mathcal{R}}(r) = \frac{1 - F_{\mathcal{T}}(r)}{T}, \tag{10}$$

where $T \triangleq E\{\mathcal{T}\}$.

Proof. We introduce the intermediate random variable $\mathcal{T}'$ to model the duration of the active demand at any uniformly-selected instant. By the law of the total expectation:

$$f_{\mathcal{R}}(r) = \int_0^{\infty} f_{\mathcal{R}|\mathcal{T}'}(r|t) f_{\mathcal{T}'}(t) dt, \tag{11}$$

where $f_{\mathcal{T}'}(t)$ is the pdf of $\mathcal{T}'$ and $f_{\mathcal{R}|\mathcal{T}'}(r|t)$ is the conditional pdf of $\mathcal{R}$ when the most recent demand is known. The probability of randomly selecting a demand is directly proportional to its duration. From this fact and the law of total probability it follows that

$$f_{\mathcal{T}'}(t) = \frac{t \cdot f_{\mathcal{T}}(t)}{\int_0^{\infty} \xi \cdot f_{\mathcal{T}}(\xi) d\xi} = \frac{t \cdot f_{\mathcal{T}}(t)}{\mathrm{E}\{\mathcal{T}\}}. \tag{12}$$

The conditional pdf $f_{\mathcal{R}|\mathcal{T}'}(r|t_i)$ yields the probability density of selecting an instant that is r time units after the start of demand d_i, given that the active demand is d_i. Since there must be no bias when selecting these instants, it is clear that $f_{\mathcal{R}|\mathcal{D}'}(r|t_i) = \frac{1}{t_i}$ if $0 \leq r < t_i$ and $f_{\mathcal{R}|\mathcal{D}'}(r|d_i) = 0$ otherwise. Combining (12) with this fact results in:

$$f_{\mathcal{R}}(r) = \int_r^{\infty} \frac{f_{\mathcal{T}}(t)}{\mathrm{E}\{\mathcal{T}\}} dt, \tag{13}$$

which directly leads to (10). □

Using Lemma III.4 we can directly calculate an expression for $f_{\mathcal{X}}(x)$, as shown in the following lemma.

Lemma III.5. *The pmf $f_{\mathcal{X}}(x)$ of the state delay of an action-interrupting network is*

$$f_{\mathcal{X}}(x) = \begin{cases} \alpha\varphi & \text{if } x = 0, \\ (1-\alpha\varphi)(1-\beta\varphi)^{x-1}\beta\varphi & \text{if } x > 0, \end{cases} \tag{14}$$

where

$$\alpha \triangleq \frac{1}{T}\int_0^\infty F_{\mathcal{Z}}(t)\left(1 - F_{\mathcal{T}}(t)\right) dt \tag{15}$$

and

$$\beta \triangleq \int_0^\infty F_{\mathcal{Z}}(t) f_{\mathcal{T}}(t) dt. \tag{16}$$

Proof. An action-interrupting network resets its action phase every time the demand changes. Hence, for any instant, the probability of operating with state delay $\mathcal{X} = 0$ is the probability of being able to find a valid state and surpassing the action time for the most recent demand. The probability of the former event is given by the maximum flexibility φ, whereas the latter event is derived as follows [18]:

$$f_{\mathcal{X}}(0) = \varphi \Pr\{\mathcal{Z} \leq \mathcal{R}\} = \varphi \int_0^\infty F_{\mathcal{Z}}(r) f_{\mathcal{R}}(r) dr, \tag{17}$$

which leads to the first case of (14) after substituting (10). If this is not the case, with probability $(1 - \alpha\varphi)$, the probability of reaching a state delay $x > 0$ is the probability of being able to find a valid solution after x unsuccessful attempts. This event follows a geometric distribution of parameter p:

$$p = \varphi \Pr\{\mathcal{Z} \leq \mathcal{T}\} = \varphi \int_0^\infty F_{\mathcal{Z}}(r) f_{\mathcal{T}}(r) dr, \tag{18}$$

which is the probability of obtaining a valid solution within an action phase that is shorter than the duration of a demand. Given the pmf of a geometrically-distributed random variable as $p(1 - p)^{x-1}$ (for $x > 0$), (14) is finally obtained. □

With an expression for $f_{\mathcal{X}}(x)$, we calculate the resulting mean readiness cost in the following theorem.

Theorem III.6. *The mean readiness cost K of an action-interrupting network is*

$$K = \alpha\varphi K_{\mathcal{X}}(0) + (1 - \alpha\varphi)\beta\varphi \widehat{K}_\beta, \tag{19}$$

where

$$\widehat{K}_\beta \triangleq \sum_{x=1}^{\infty} K_{\mathcal{X}}(x)(1 - \beta\varphi)^{x-1}. \tag{20}$$

Proof. Eq. (19) is the result of combining (9) and (14). □

The expression in Theorem III.6 allows us to calculate the mean readiness cost of an adaptive system from its demand duration distribution, action time distribution, readiness degradation function, and maximum flexibility. In order to find out if a network is profitable, the following corollary can be used.

Corollary III.6.1. *A necessary condition for a flexible network to be profitable is*

$$\frac{\alpha}{(\alpha\varphi - 1)\beta} < \frac{\widehat{K}_\beta}{K_{\mathcal{X}}(0)}. \tag{21}$$

given that $K_{\mathcal{X}}(0) < 0$.

Proof. A profitable network must fulfill $K < 0$, which implies that $K_{\mathcal{X}}(0) < 0$ (Lemma III.2). After applying these relations in (19), we reach (21). □

Corollary III.6.1 provides us with a simple method to rule out non-profitable network configurations. The left side of (21) reflects the frequency of demand changes and the swiftness and effectiveness of the adaptation, whereas the right side is influenced by the quality of the solutions. As a result, it delimits a border for finding profitable configurations within the speed-quality tradeoff. An example application of Theorem III.6 and Corollary III.6.1 is presented in Sec. IV-B.

C. Proaction cost

The proaction cost $c^P(\widetilde{d}_i)$ of a flexible network can be formally defined as a function mapping a demand change to a cost value:

$$c^P(\widetilde{d}_i) : \mathbb{D}^2 \mapsto \mathbb{R}^+. \tag{22}$$

The exclusive dependency on $\widetilde{d}_i$ implies that this cost is present every time there is a demand change, even if there is no eventual state change. We identify two factors contributing to the proaction cost. On the one hand, a demand change may imply an instantaneous time-independent cost C_0^P, for instance resulting from the activation of new capabilities to solve the adaptation problem. On the other hand, while the adaptation problem is being solved, additional resources (CPU, memory, etc.) are consumed during the proaction phase, incurring in a cost of C_z^P monetary units per time unit. As a result, we can express the proaction cost as a function of the proaction time of an action-interrupting network as follows:

$$c^P(\widetilde{d}_i) \triangleq \frac{1}{t_{i+1}}\left(C_0^P + C_z^P \cdot \min(z_i^P, t_{i+1})\right), \tag{23}$$

where the minimum operator guarantees that the proaction phase is stopped if the demand changes and the term $\frac{1}{t_{i+1}}$ normalizes the cost to the duration of the demand. From (23) and the sequence of proaction times $\{\mathcal{Z}_i^P\}$ we define a new stochastic process $\{\mathcal{C}_i^P\}_{i\in\mathbb{Z}}$ to model the sequence of proaction costs, such that:

$$\mathcal{C}_i^P = \frac{1}{\mathcal{T}_{i+1}}\left(C_0^P + C_z^P \cdot \min(\mathcal{Z}_i^P, \mathcal{T}_{i+1})\right). \tag{24}$$

The mean of the variable above is presented in the following.

Theorem III.7. *The mean proaction cost C^P of an action-interrupting network is:*

$$C^P = \frac{C_0^P}{T} + \left(\frac{\beta^P Z^P}{T} + 1 - \beta^P\right) C_z^P, \tag{25}$$

where

$$\beta^P \triangleq \int_0^\infty F_{\mathcal{Z}^P}(t) f_{\mathcal{T}}(t) dt, \tag{26}$$

and $Z^P \triangleq E\{\mathcal{Z}^P\}$.

Proof. After applying the expectation operator to (24), we need to calculate $\mathrm{E}\{\min(\mathcal{Z}_i^P, \mathcal{T}_{i+1})\}$. The random variable within the brackets takes the same values as $\mathcal{Z}_i^P$ when $\mathcal{Z}_i^P \leq \mathcal{T}_{i+1}$. From stationarity and the law of total expectation:

$$\begin{aligned}\mathrm{E}\{\min(\mathcal{Z}^P, \mathcal{T})\} = \Pr\{\mathcal{Z}^P \leq \mathcal{T}\} \cdot Z^P \\ + (1 - \Pr\{\mathcal{Z}^P \leq \mathcal{T}\}) \cdot T\end{aligned} \tag{27}$$

The probability $\Pr\{\mathcal{Z}^P \leq \mathcal{T}\}$ is derived in the same way as (16) to yield (26). $\square$

In Sec. IV-C, we show an example network where we apply Theorem III.7 to find out the optimal number of CPU cores to be used during the proaction phase.

Corollary III.7.1. *A necessary condition for an action-interrupting network to be profitable is*

$$C_z^P < \frac{TK + C_0^P}{(\beta^P - 1)T - \beta^P Z^P}. \tag{28}$$

Proof. This relation follows directly from the fact that a profitable network must fulfill $K + C^P < 0$. $\square$

Corollary III.7.1 provides us with an upper bound on the maximum number of additional resources that a flexible network is allowed to utilize in order to cope with demand changes before it turns unprofitable.

D. Reaction cost

The reaction cost $c^R(\widetilde{s}_i)$ reflects the effort of performing the state change required for an adaptation, after this has been selected in the proaction phase. Therefore, we define it as a function of the state change $\widetilde{s}_i$:

$$c^R(\widetilde{s}_i) : \mathbb{S}^2 \mapsto \mathbb{R}^+. \tag{29}$$

Note that $c^R(\widetilde{s}_i) = 0$ if the demand change $\widetilde{d}_i$ results in no adaptation, that is, if $s_{i+1} = s_i$. This can happen either if s_i is already optimal, or if the adaptation algorithm could not find a better state which satisfying d_j.

Based on the same rationale as with the proaction cost, we identify two factors contributing to the reaction cost. First, an instantaneous time-independent cost C_0^R models the activation of state-changing procedures (such as memory allocation for virtual migrations [19], for instance). Second, we define a constant rate of C_z^R monetary units per time unit to characterize the usage of additional resources when changing the network state. Consequently, we formulate the reaction cost as:

$$c^R(\widetilde{s}_i) \triangleq \frac{1}{t_{i+1}} \left(C_0^R + C_z^R \cdot \min(z_i^R, t_{i+1} - z_i^P) \right) \tag{30}$$

whenever $t_{i+1} \leq z_i^P$ and the demand is satisfiable. Otherwise, $c^R(\widetilde{s}_i) = 0$ as no new state has been generated. We define the stochastic process $\{\mathcal{C}_i^R\}_{i \in \mathbb{Z}}$ to model a time-ordered sequence of reaction costs as:

$$\mathcal{C}_i^R = \frac{1}{\mathcal{T}_{i+1}} \left(C_0^R + C_z^R \cdot \min(\mathcal{Z}_i^R, \mathcal{T}_{i+1} - \mathcal{Z}_i^P) \right), \tag{31}$$

for every index i whenever $\mathcal{T}_{i+1} \leq \mathcal{Z}_i^P$ and $\mathcal{C}_i^R = 0$ otherwise.

Theorem III.8. *The mean reaction cost C^R of an action-interrupting network is:*

$$E\{\mathcal{C}^R\} = \frac{\varphi \beta^P}{T} \left(C_0^R + C_z^R \left(\beta Z (1 - \beta) T - Z^P \right) \right). \tag{32}$$

Proof. The equality (31) occurs when $\mathcal{Z}_i^P \leq \mathcal{T}_{i+1}$ with probability β^P. By the law of total expectation, we just need to figure out the value of $\mathrm{E}\{\min(\mathcal{Z}_i^R, \mathcal{T}_{i+1} - \mathcal{Z}_i^P)\}$. The random variable within the brackets takes the same values as $\mathcal{Z}_i^R$ when $\mathcal{Z}_i \leq \mathcal{T}_{i+1}$, that is, with probability β. After some straightforward algebra, (32) is obtained. $\square$

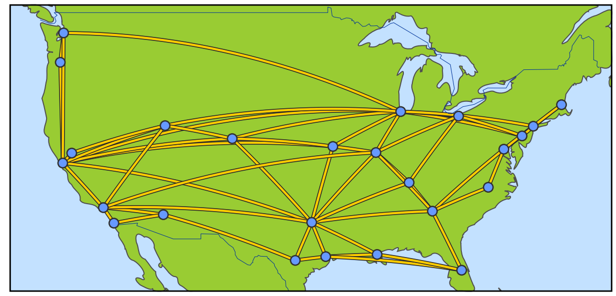

Fig. 3: Topology of the example network implementing random flow requests. The topology is that of AT&T North America [20] taken from the Internet Topology Zoo [21].

IV. Application Example

In this section, we show an example application of the cost model. Namely, we present a network design problem, lay out several alternatives to solve it, and use the presented cost model to figure out the best implementation options.

A. Network description

We consider a communication network with the topology shown in Fig. 3. The purpose of the network is to embed flow requests between any two nodes, that is, to configure its links so that connection between a pair of nodes can be established. The network operator wants to provide this service with the minimum operational cost while ensuring that flows are implemented on single paths, i. e., without fractional flows.

Each flow request consists of a source-destination pair and a required throughput. We can thus define the demand d at any time as the set of currently requested flows. As a result, every time a new flow is requested or finishes, the demand changes. The instantaneous network state s is defined by the state of each link (active or inactive) and its assignment to an active flow, if any. When a link is active, multiple flows can be assigned to it as long as the capacity of the link is not exceeded. For simplicity, we set all link capacities to 1 and the throughput requested by each flow is uniformly chosen between 0 and 1. We consider that a demand is satisfied when the network state allows the embedding of all flows contained in the demand.

The duration of the demands $\mathcal{T}$ (in seconds) follows a Pareto Type II distribution (also called Lomax distribution [22]):

$$F_{\mathcal{T}}(t) = 1 - \left(1 + \frac{t}{\lambda}\right)^{\sigma}, \tag{33}$$

for $t \geq 0$ with parameters $\lambda = 10$ and $\sigma = 2.25$ such that the mean demand duration is $T = \frac{\lambda}{\sigma - 1} = 8$ seconds. We select this distribution since it is frequently observed in the interarrival time between internet bursts, file sizes, transfer times, etc [23]. In addition, we choose a low demand duration, comparable to the action time, so that it would be unclear whether a network can operate profitably in this situation when using conventional cost models. Nonetheless, other parameters or distributions, such as the exponential distribution, can be used without affecting the effectiveness of the model.

The problem of providing the intended connectivity can be formulated as an instance of the *integer* min-cost multicommodity flow problem [24]. This problem is known to be NP-Hard [25], thus the network operator relies on an approximation algorithm rather than on an exact approach. In our example, the operator uses a genetic algorithm as the adaptation algorithm [26]. In a nutshell, the operation of the genetic algorithm is as follows [27]. First, a certain number of random solutions, called the *population size* π, are generated. Each solution contains a flag per link indicating if this link is active or not. Then each solution is evaluated to assess how close it is to satisfy the current demand and how many links it uses. After all solutions have been evaluated, the worst ones are discarded whereas the best ones are kept for the next generation. These are then combined to each other and randomly modified to produce the next generation. These steps are repeated until a convergence criterion is satisfied, which in our case is the absence of improvements after 25 generations.

B. Selection of a profitable population size

Given the high number of parameters in a genetic algorithm, the operator is interested in finding the right ones so that the network is profitable. For example, they want to select the population size π. Selecting the right π is not trivial, since it affects the speed-accuracy trade-off of the algorithm. On the one hand, a large π increases the probability of eventually finding an optimal solution with minimum readiness cost. On the other hand, the higher π, the more solutions have to be evaluated, which increases proaction time and cost.

The readiness cost k_j at any point is the combination of three factors. First, subscribed users provide a constant revenue of 71 monetary units per second (mu/s). Second, active links have a cost of 11 mu/s, whereas inactive links do not cost anything. Finally, whenever a requested flow is not being satisfied, the operator has to pay a compensation of 10 mu/s to each affected user. Thus, the average readiness cost of state s and demand d is:

$$k(s,d) = -71 + 11l(s) + 10v(s,d) \text{ mu/s}, \tag{34}$$

where $l(s)$ is the number of used links in state s and $v(s,d)$ is the number of unsatisfied flows for demand d and state s.

The operator is considering to use a population size in the set $\Pi = \langle 250, 750, 1250, 1750, 2250, 2750\rangle$. After evaluating the performance of the genetic algorithm in a dedicated simulator, we observe that the action time can be modeled by a uniform distribution such that $\mathcal{Z} \sim \mathcal{U}(0, \widehat{Z})$, where $\widehat{Z} = 0.016 \cdot \pi$. We also measure the maximum flexibilities $\Phi = \langle 0.35, 0.55, 0.64, 0.7, 0.73, 0.75\rangle$ for each π in the same order as Π, representing the frequency of demand-satisfying solutions. Finally, we measure the readiness degradation function $K_{\mathcal{X}}(x)$ for state delays $0 \le x \le 30$ for each population size π, which is shown in Fig. 4. We observe that the mean

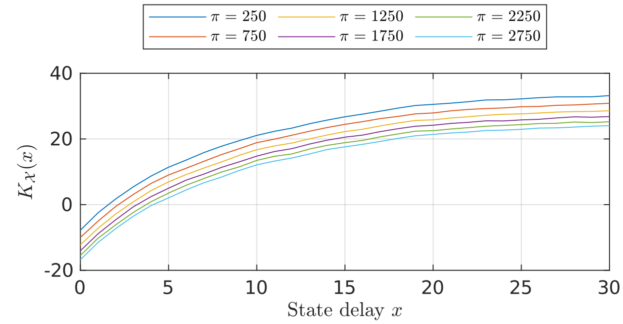

Fig. 4: Readiness degradation function $K_{\mathcal{X}}(x)$ of a network employing a genetic algorithm to implement integer multicommodity flows for six different population sizes.

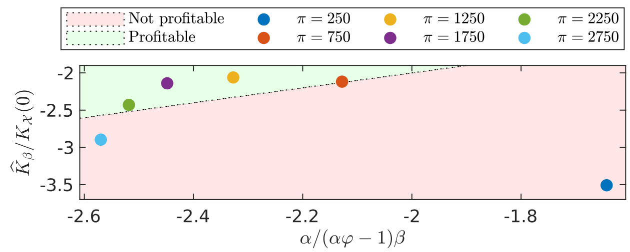

Fig. 5: Graphical representation of inequality (21) for testing the profitability of different population sizes.

readiness cost steadily increases with the state delay, as a result of higher compensations due to unsatisfied flows and unnecessarily active links. It is also clear that the evolution of the cost for the different population sizes is very similar, although low population sizes lead to higher link usage, which increases the cost.

From the above measurements and the distributions of $\mathcal{T}$ and $\mathcal{Z}$, we are able to compute parameters α and β as shown in (15) and (16). With this and the readiness degradation function $K_{\mathcal{X}}(x)$, we can apply Corollary III.6.1 to figure out if any of the considered population sizes may lead to a profitable network. A graphical representation of the result is shown in Fig. 5, where left and right sides of inequality (21) are depicted as horizontal and vertical axes, respectively. We observe that three of the considered population sizes (π = 1250, 1750, 2250) lie on the profitable region. An interesting behavior is captured by the model, as those populations that are lower than 1250 or greater than 2250 lead to unprofitable networks. The explanation is that, when the population is small, the quality of the states yielded by the genetic algorithm is not good enough to properly address the demands. Conversely, when the population is large, the action time is so high that the network cannot properly cope with frequent demand changes.

C. Optimal parallelization level

Let us now consider that the network operator has the ability to dedicate multiple CPU cores to solving the adaptation problem in the proaction phase. Increasing the number of cores reduces the proaction time, thus decreasing the state delay, which may lead to higher revenue. Nevertheless, utilizing more

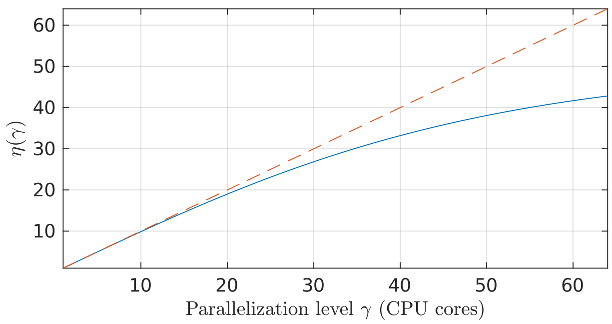

Fig. 6: Reduction factor of the proaction time for different parallelization levels. The dashed line has unit slope.

cores also increases the proaction cost, which may counter the revenue increase and result in higher total cost. In order to find out the optimal level of parallelization, we can combine Theorems III.6 and III.7 to predict the evolution of readiness and proaction costs as the number of cores assigned to the adaptation algorithm increases.

Let us denote the parallelization level, i. e., the number of additional CPU cores used in the proaction phase, as γ. For simplicity, let us assume that the reaction cost and time are negligible so that $\mathcal{Z}^{\mathcal{P}} \approx \mathcal{Z}$, hence $\mathcal{Z}^{\mathcal{P}} \sim \mathcal{U}(0, \widehat{Z})$. Clearly, the value of $\widehat{Z}$ is a decreasing function of γ. Namely, we define it as $\widehat{Z} = \frac{\widehat{Z}_0}{\eta(\gamma)}$, where $\widehat{Z}_0$ is the maximum proaction time with a single core and $\eta(\gamma)$ is the time reduction factor for γ cores. Ideally, $\eta(\gamma) = \gamma$ if the load can be perfectly shared among all cores. Nonetheless, in real scenarios it is observed that $\eta(\gamma)$ grows linearly at first, but eventually saturates due to imperfections in load division [28], [29]. To capture this, we use the $\eta(\gamma)$ depicted in Fig. 6.

Regarding the proaction cost components, let $C_0^P = 0$ (no additional cost for starting the proaction cost) and $C_z^P = 1 \cdot \gamma$ mu/s, where γ is the number of assigned CPU cores, that is, the parallelization level. This value of C_z^P means that the network consumes 1 mu/s per used CPU core during the proaction phase, in addition to the readiness cost. Finally, based on the previous results, we select a population size of $\pi = 1750$, which implies a maximum flexibility of $\varphi = 0.7$ and $\widehat{Z}_0 = 28$.

We can calculate the relationship between the mean proaction cost C^P and the parallelization level γ by feeding the aforementioned expressions into (25), in Theorem III.7. The result is shown in Fig. 7, where an interesting behavior can be observed. Up to around $\gamma = 8$, the proaction cost increases rapidly, since the duration of the proaction phase is limited by the demand duration. Indeed, when $\gamma = 1$, the average demand duration is $T = 8$ s, whereas the mean proaction time is, $Z^P = \frac{\widehat{Z}_0}{2} = 14$ s. As a consequence, the network is almost always in the proaction phase, and thus increasing γ only increases the proaction cost without affecting the duration of the proaction phases. Nevertheless, as γ grows, eventually the proaction time becomes lower than the demand duration, thus allowing the network to leave the proaction phase and leading to a less steep cost increase.

Finally, in Fig. 8 we show the combined readiness and

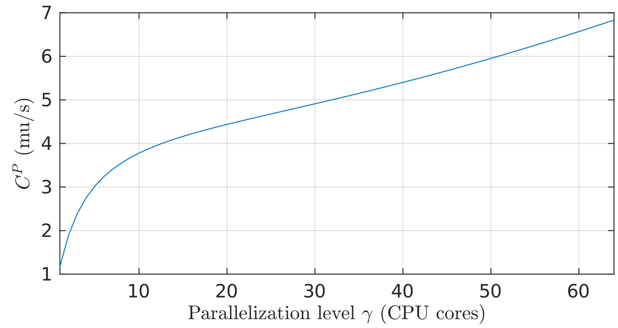

Fig. 7: Mean proaction cost C^P as a function of the parallelization level.

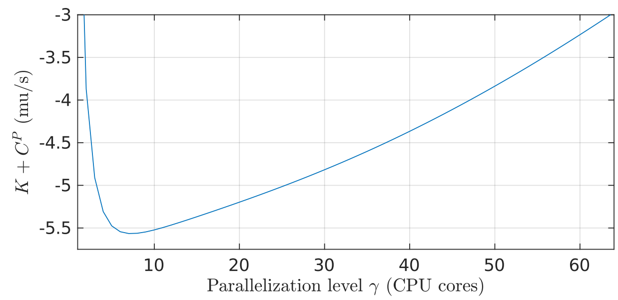

Fig. 8: Combined mean readiness and proaction costs $K + C^P$ as a function of the parallelization level.

proaction cost for this scenario, which can be achieved via Theorems III.6 and III.7. We clearly observe a minimum point at $\gamma = 7$ cores, which is thus the optimal parallelization level. Before this value the proaction cost is lower, but the network cannot cope with demand changes fast enough, resulting high readiness cost due to large state delays. For $\gamma > 7$ the readiness approaches its minimum value but the proaction cost increases, resulting in higher combined cost.

V. Conclusion

Current communication networks need to be able to easily adapt to a changing environment in order to face the evergrowing demands and the myriad of different network applications. In order to accomplish this, the concept of network softwarization proposes to replace hardware equipment with software entities. This facilitates network reconfiguration, allowing faster and less costly adaptations. However, these adaptations have multiple cost factors associated with them, which are difficult to capture using conventional models.

In this paper, a comprehensive cost model for a flexible network is proposed. The mathematical tools designed for flexibility analysis are used and extended to provide a deep understanding of all the factors affecting the cost. We identify three main cost components related to adaptations: readiness, proaction, and reaction cost, which have to be combined to calculate the total cost. We provide expressions and relationships that can be used to accurately predict cost and take design decisions. Finally, we apply the cost model to realistic examples to show how this can be done.

References

[1] Cisco, "Prepare to succeed with the internet of things," White Paper, 2017.

[2] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: Applications, challenges, and opportunities with China perspective," *IEEE Internet of Things journal*, vol. 1, no. 4, pp. 349–359, 2014.

[3] NTT DOCOMO, "5G evolution and 6G," White Paper, 1 2020.

[4] A. Martínez Alba and W. Kellerer, "A dynamic functional split in 5G radio access networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[5] B. Casemore, "How network disaggregation facilitates datacenter and it modernization," IDC, White Paper, June 2018.

[6] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.

[7] W. Kellerer, A. Basta, P. Babarczi, A. Blenk, M. He, M. Klugel, and A. Martínez Alba, "How to measure network flexibility? a proposal for evaluating softwarized networks," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 186–192, 2018.

[8] M. Bettner, S. Haka, J. Williams, and J. Carcello, *Financial & Managerial Accounting*. McGraw-Hill Education, 2017.

[9] R. Huelsermann, M. Gunkel, C. Meusburger, and D. A. Schupke, "Cost modeling and evaluation of capital expenditures in optical multilayer networks," *Journal of Optical Networking*, vol. 7, no. 9, pp. 814–833, 2008.

[10] J. Zander, "On the cost structure of future wideband wireless access," in *1997 IEEE 47th Vehicular Technology Conference. Technology in Motion*, vol. 3. IEEE, 1997, pp. 1773–1776.

[11] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[12] J. L. Higle and S. Sen, "Stochastic decomposition: An algorithm for two-stage linear programs with recourse," *Mathematics of operations research*, vol. 16, no. 3, pp. 650–669, 1991.

[13] M. Klügel, M. He, W. Kellerer, and P. Babarczi, "A mathematical measure for flexibility in communication networks," in *2019 IFIP Networking Conference (IFIP Networking)*. IEEE, 2019, pp. 1–9.

[14] P. Babarczi, M. Klügel, A. Martínez Alba, M. He, J. Zerwas, P. Kalmbach, A. Blenk, and W. Kellerer, "A mathematical framework for measuring network flexibility," *Computer Communications*, vol. 164, pp. 13–24, 2020, Special Issue on IFIP Networking 2019 Conference.

[15] L. Zhang, W. Wu, and D. Wang, "Time dependent pricing in wireless data networks: Flat-rate vs. usage-based schemes," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 700–708.

[16] G. Kesidis, A. Das, and G. de Veciana, "On flat-rate and usage-based pricing for tiered commodity internet services," in *2008 42nd Annual Conference on Information Sciences and Systems*. IEEE, 2008, pp. 304–308.

[17] P. Billingsley, *Probability and Measure*, ser. Wiley Series in Probability and Statistics. Wiley, 1995.

[18] R. Durrett, *Probability: Theory and Examples*, ser. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2010.

[19] A. Martínez Alba, J. H. Gómez Velásquez, and W. Kellerer, "An adaptive functional split in 5G networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 410–416.

[20] ATT North America. Next-generation IP MPLS backbone. [Online]. Available: https://www.att.com/Common/merger/files/pdf/wired-network/Domestic_0C-768_Network.pdf

[21] The Internet Topology Zoo. Accessed on 01/14/2021. [Online]. Available: http://www.topology-zoo.org/

[22] K. S. Lomax, "Business failures: Another example of the analysis of failure data," *Journal of the American Statistical Association*, vol. 49, no. 268, pp. 847–852, 1954.

[23] A. B. Downey, "Lognormal and pareto distributions in the internet," *Computer Communications*, vol. 28, no. 7, pp. 790–801, 2005.

[24] J. Tomlin, "Minimum-cost multicommodity network flows," *Operations Research*, vol. 14, no. 1, pp. 45–51, 1966.

[25] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multi-commodity flow problems," in *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*. IEEE, 1975, pp. 184–193.

[26] N. Farrugia, J. A. Briffa, and V. Buttigieg, "Solving the multi-commodity flow problem using a multi-objective genetic algorithm," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 2816–2823.

[27] K. Man, K. Tang, and S. Kwong, *Genetic Algorithms: Concepts and Designs*, ser. Advanced Textbooks in Control and Signal Processing. Springer London, 2001.

[28] G. Glockner, "Parallel and distributed optimization with gurobi optimizer," 2015.

[29] D. Bergman, A. A. Cire, A. Sabharwal, H. Samulowitz, V. Saraswat, and W.-J. van Hoeve, "Parallel combinatorial optimization with decision diagrams," in *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*. Springer, 2014, pp. 351–367.