End-to-End Quality-of-Service Assurance with Autonomous Systems: 5G/6G Case Study

Van Sy Mai
CTL, NIST
Gaithersburg, MD USA
vansy.mai@nist.gov

Richard J. La
ITL, NIST and
ECE & ISR, Univ. of MD
Gaithersburg, MD USA
richard.la@nist.gov

Tao Zhang
CTL, NIST
Gaithersburg, MD USA
tao.zhang@nist.gov

Abdella Battou
CTL, NIST
Gaithersburg, MD USA
abdella.battou@nist.gov

Abstract—**Providing differentiated services to meet the unique requirements of different use cases is a major goal of the fifth generation (5G) telecommunication networks and will be even more critical for future 6G systems. Fulfilling this goal requires the ability to assure quality of service (QoS) end to end (E2E), which remains a challenge. A key factor that makes E2E QoS assurance difficult in a telecommunication system is that access networks (ANs) and core networks (CNs) manage their resources autonomously. So far, few results have been available that can ensure E2E QoS over autonomously managed ANs and CNs. Existing techniques rely predominately on each subsystem to meet static local QoS budgets with no recourse in case any subsystem fails to meet its local budgets and, hence will have difficulty delivering E2E assurance. Moreover, most existing distributed optimization techniques that can be applied to assure E2E QoS over autonomous subsystems require the subsystems to exchange sensitive information such as their local decision variables. This paper presents a novel framework and a distributed algorithm that can enable ANs and CNs to autonomously "cooperate" with each other to dynamically negotiate their local QoS budgets and to collectively meet E2E QoS goals by sharing only their estimates of the global constraint functions, without disclosing their local decision variables. We prove that this new distributed algorithm converges to an optimal solution almost surely, and also present numerical results to demonstrate that the convergence occurs quickly even with measurement noise.**

I. Introduction

A major goal of the fifth generation (5G) telecommunication networks is to provide differentiated (or even optimized) services to meet different use cases' unique requirements, e.g., on performance, availability, and security [1]. The ability to provide such differentiated support is a prerequisite for many IoT (Internet of Things) applications such as automated manufacturing, vehicle-to-vehicle communications, remote operation (teleoperation) of ground vehicles and drones, and telesurgery. It represents a fundamental advancement over the previous generations of networks that had focused on providing one-size-fits-all network connectivity for all applications.

To enable differentiated services, 5G shifted from a communication-centric architecture to a service-centric architecture that separated network and service control functions from the physical infrastructure, further modularized these functions, and defined their interactions in terms of providing and consuming services. This service-centric architecture enables network slicing that creates end-to-end (E2E) logical networks (network slices) dedicated for specific services, forming a foundation for delivering differentiated services. Current 5G specifications further describe a framework and its enabling protocols for collecting network performance data required to support service assurance [3]–[7].

Despite recent advances, fundamental challenges remain to be addressed before the promise of differentiated services can be fully realized in 5G and future generations of telecommunication networks [19]. One such challenge is to assure E2E quality of service (QoS). Such assurance is essential to, for example, many delay-sensitive applications [24], [25] and will be more critical for future 6G systems that will support more use cases and more stringent requirements [15], [19].

A key factor that makes E2E service assurance difficult in telecommunication networks is the fact that access networks (ANs) and core networks (CNs) manage their resources autonomously. While such autonomous management simplifies the tasks of designing, operating, and evolving a large system, limited coordination among ANs and CNs on resource management also leaves E2E service assurance more challenging [20].

To achieve E2E QoS assurance over autonomously managed ANs and CNs, current 5G specifications decompose the E2E Key Performance Indicator (KPI) targets (e.g., packet delay, packet drop ratio) into KPI budgets for each CN and AN, and rely on each of them to autonomously manage its resources to satisfy its allocated KPI budgets. For example, 5G specifications decompose the E2E Packet Delay Budgets (PDBs) into CN PDBs and AN PDBs, with the CN PDBs for standardized QoS flows predefined in the specifications and to be statically configured in each CN [2]. A network operator will then need to determine the AN PDBs for each AN, which is done statically today.

Static KPI budget allocations cannot effectively respond to dynamic network conditions or account for the varying characteristics of different networks (e.g., different network capabilities, capacities, and supported services). Also, since an AN or CN may violate its KPI budgets due to random events such as network congestion, network operators will need ways to ensure that such violations will not lead to intolerable violations of E2E KPI requirements.

Recognizing the limitations of static KPI budget allocations, 5G permits non-standard QoS flows to support dynamic PDBs and allows the dynamic PDBs to be signaled between ANs and CNs. However, current 5G specifications do not prescribe how dynamic PDBs for each AN or CN should be determined or enforced, and how E2E PDB requirements can be assured in case any autonomous network along an E2E path violates its allocated PDB budgets.

The past few years have also seen growing efforts in developing resource allocation mechanisms for assuring E2E QoS for 5G and beyond systems. Here, we summarize only the most closely related studies that focused on E2E QoS rather than QoS for RANs or CNs alone. We emphasize that this is not meant to be an exhaustive list. In [14], the 5G QoS architecture is extended to incorporate the Framework for Abstraction and Control of Traffic Engineered Networks (ACTN), which was developed by the IETF (IETF RFC 8453) to manage large multi-domain networks. This extended architecture calls for the resource controllers in different technology or administrative domains (e.g., ANs and CNs) to be orchestrated by a Multi-Domain Service Coordinator function to achieve E2E service assurance. But, it does not describe how such orchestration can be accomplished. In [22], a high-level architecture for 5G service assurance is described. It focuses on monitoring E2E KPIs rather than providing mechanisms for achieving E2E service assurance. A dynamic resource slicing method aimed at E2E service assurance is proposed in [23], and it treats resource slicing in the radio ANs and the CN independently. A technique that combines user device QoS provisioning with network QoS provisioning is proposed in [8]. It treats QoS provisioning in ANs and CNs as independent tasks and relies on an unrealistic assumption that the KPI budgets for each network node are fixed and known. In [11], path configuration methods are proposed for ensuring E2E QoS in Software-Defined Networks (SDNs). SDNs are expected to be widely used in 5G systems. But, the path configuration methods in [11] require centralized provisioning and control of E2E network paths, which is impractical in telecommunication systems where ANs and CNs are expected to continue to be managed separately.

In order to assure E2E service quality while allowing the ANs and the CNs to control their resources independently, these autonomous networks should be able to coordinate with each other to meet E2E KPI requirements. To this end, we propose an optimization-based framework in which the E2E KPI requirements are captured by *global* constraints. Based on this framework, we design a new *distributed* algorithm; existing distributed optimization algorithms require the subsystems to maintain and exchange their local decision variables with each other. This can result in heavy signaling overheads, increase technology inter-dependency among ANs and CNs, or reveal how different subsystems manage their internal resources, and thus are impractical or undesirable in telecommunication systems [9], [12], [21].

In contrast, our algorithm only needs the subsystems to exchange their *estimates of the global constraint functions*. When the subsystems update their local decision variables using the proposed algorithm, they converge to an optimal point with probability 1. We present numerical results to demonstrate that the proposed algorithm quickly reduces the cost of the system to near the optimal value even with measurement noise. Finally, we illustrate that each optimal point gives rise to a stationary state Nash equilibrium of an associated state-based potential game.

Notation: We denote set of nonnegative integers by $\mathbb{N}$. Unless stated otherwise, all vectors are column vectors and $||\cdot||_2$ denotes the ℓ_2 norm. Given a vector $\mathbf{x}$, we denote the kth element by x_k. The vector of zeros (resp. ones) of appropriate dimension is denoted by $\mathbf{0}$ (resp. $\mathbf{1}$). Given a closed convex set $\mathcal{S}$ and a vector $\mathbf{x}$, $P_{\mathcal{S}}(\mathbf{x})$ denotes the Euclidean projection of $\mathbf{x}$ onto $\mathcal{S}$.

II. Proposed Optimization Framework

In the rest of the paper, we refer to an AN or a CN as an agent, and let $\mathcal{A}$ be the set of N agents. The problem of ensuring E2E QoS over autonomously managed networks can be formulated as a distributed optimization problem with both *local* and *global* constraints. Let $\mathbf{y}_i$ be the local decision variables of agent $i \in \mathcal{A}$, which are used by agent i to manage its network resources. The local constraints of agent i restrict $\mathbf{y}_i$ independently of the decision variables chosen by other agents. We call the set of $\mathbf{y}_i$ that satisfies agent i's local constraints its local constraint set and denote it by $\mathcal{G}_i$. Let $\mathbf{y} := (\mathbf{y}_i : i \in \mathcal{A})$ be the vector comprising the decision variables of all agents. The set of $\mathbf{y}$ that satisfies the local constraints of all agents is denoted by $\mathcal{G} := \prod_{i\in\mathcal{A}} \mathcal{G}_i$.

We capture the E2E QoS guarantees, which the agents must meet collectively, using K global constraint functions $g_1, \ldots, g_K$. The global constraints depend on the decision variables of more than one agent and, hence, require their coordination.

If (i) the local objective functions of individual agents, (ii) local constraint sets $\mathcal{G}_i$, $i \in \mathcal{A}$, and (iii) global constraint functions $g_1, \ldots, g_K$ are known, we can formulate the problem of optimizing the overall network performance subject to the E2E QoS requirements as a constrained optimization problem of the form given below, where (a) the objective function is separable, and (b) global (inequality) constraints couple the decision variables of agents:

$$\min_{\mathbf{y}\in\mathcal{G}} \quad \sum_{i\in\mathcal{A}} \phi_i(\mathbf{y}_i) \tag{1a}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{y}) := \sum_{i\in\mathcal{A}} \mathbf{g}_i(\mathbf{y}_i) \le \mathbf{0}, \tag{1b}$$

where ϕ_i denotes the local objective function that agent i aims to minimize using its decision variables $\mathbf{y}_i$. Here, $\mathbf{g} := (g_1, \ldots, g_K)$ is the vector function consisting of K global constraint functions. In our setting, each agent i knows $\mathbf{g}_i$, which represents its contributions to global constraint functions, but not $\mathbf{g}_j$, $j \in \mathcal{A} \setminus \{i\}$.

A. Example: End-to-end Delay Constraints

Consider a network consisting of N domains or autonomous systems, which are the agents in our framework, and denote the set of agents by $\mathcal{A}$. Suppose that the network provides n_S different service classes and $\mathcal{S}$ is the set of service classes. Let $\mathcal{F}$ be the set of n_F flows that need to be supported. Each flow $f \in \mathcal{F}$ is of service class $s_f \in \mathcal{S}$. Thus, the set $\mathcal{F}$ can be partitioned into $\{\mathcal{F}^s : s \in \mathcal{S}\}$, where $\mathcal{F}^s$ is the set of service class s flows in $\mathcal{F}$ with $n^s := |\mathcal{F}^s|$, $s \in \mathcal{S}$. Each flow f traverses a subset of agents in $\mathcal{A}_f$. We use an $n^s \times N$ matrix $\mathbf{R}^s = [R^s_{f,a} : f \in \mathcal{F}^s, a \in \mathcal{A}]$ to describe the routes taken by the flows of service class s, where $R^s_{f,a} = 1$ if flow f traverses agent a, and $R^s_{f,a} = 0$ otherwise.

a) Problem for individual agents: Let $\mathcal{N}^a$ be the set of egress/ingress nodes in agent a and $m^a := |\mathcal{N}^a|$. An agent a's flow b refers to a triple $(s, p_i, p_e) \in \mathcal{S} \times \mathcal{N}^a \times \mathcal{N}^a$ with the understanding that it carries service class s traffic that enters and leaves agent a at p_i (ingress point) and p_e (egress point), respectively. We denote the set of flows of service class s in agent a by $\tilde{\mathcal{F}}^a_s$, and let $\tilde{\mathcal{F}}^a := \cup_{s\ \in \mathcal{A}} \tilde{\mathcal{F}}^a_s$.

For each agent $a \in \mathcal{A}$, there is a fixed traffic tensor that describes the amount of traffic that needs to be transported for each ingress-egress pair and each service class. We denote this traffic tensor by $\mathbf{T}^a$, which is an $n_s \times m^a \times m^a$ tensor; the element T^a_{s,p_i,p_e} denotes the demand of flow $b = (s, p_i, p_e)$, i.e., it is the amount of traffic belonging to service class $s \in \mathcal{S}$, which needs to be transported from ingress node $p_i \in \mathcal{N}^a$ to egress node $p_e \in \mathcal{N}^a$. We use $I(b)$ and $E(b)$ to denote the ingress and egress node, respectively, of flow b.

Let $\mathcal{L}^a$ be the set of (directed) links in agent a. Each link $\ell \in \mathcal{L}^a$ has a finite capacity c^a_ℓ, and $\mathbf{c}^a := (c^a_\ell : \ell \in \mathcal{L}^a)$. Each flow $b \in \tilde{\mathcal{F}}^a$ is supported by a set of routes in $\mathcal{R}^a_b$ connecting $I(b)$ to $E(b)$, and a route $r \in \mathcal{R}^a_b$ utilizes a subset of links in agent a, which we denote by $\mathcal{L}^a(r)$ with a little abuse of notation.[1] For each $s \in \mathcal{S}$, we define $\mathcal{P}^a_s := \cup_{b \in \tilde{\mathcal{F}}^a_s} \mathcal{R}^a_b$ and a routing matrix $\mathbf{R}^a_s$, which is a $|\mathcal{P}^a_s| \times |\mathcal{L}^a|$ matrix; the element $R^a_{r,\ell} = 1$ if $\ell \in \mathcal{L}^a(r)$ and $R^a_{r,\ell} = 0$ otherwise.

The variable $x^a_{b,r}, r \in \mathcal{R}^a_b$ for some flow $b \in \tilde{\mathcal{F}}^a$, denotes the amount of flow b traffic traversing route r. Define $\mathbf{x}^a = (x^a_{b,r} : b \in \tilde{\mathcal{F}}^a$ and $r \in \mathcal{R}^a_b)$ to be the *local* decision variables of agent a. These local decision variables must be nonnegative and support the given traffic tensor: for every $b \in \mathcal{F}^a$,

$$x^a_{b,r} \geq 0 \text{ for all } r \in \mathcal{R}^a_b \quad \text{and} \quad \textstyle\sum_{r \in \mathcal{R}^a_b} x^a_{b,r} = T^a_b.$$

We use $\mathcal{G}_a$ to denote the set of agent a's decision variables that satisfy these conditions.

The goal of agent a is to minimize its own *local* cost given by $C^a(\mathbf{x}^a)$. This local cost $C^a(\mathbf{x}^a)$ can be, for example, the (weighted) average delay experienced by the traffic traversing agent a or the costs associated with packet losses suffered in its network. Oftentimes, this cost includes a summable cost, i.e., the sum of the costs of individual links, which can be written as $\sum_{s \in \mathcal{S}} \left(\sum_{\ell \in \mathcal{L}^a} c_{s,\ell}(\mathbf{x}^a)\right)$, where $c_{s,\ell}(\mathbf{x}^a)$ is the cost of class s traffic on link ℓ as a function of $\mathbf{x}^a$.

b) E2E delay requirements: The traffic of service class s cannot tolerate (average) E2E delays larger than $d^*(s)$. We handle these E2E delay constraints using the following *global* constraints: suppose that, given local decision variables $\mathbf{x}^a$ of agent a, the delay experienced by service class s traffic on link ℓ is given by $D^a_{s,\ell}(\mathbf{x}^a)$. Then, the delays experienced by flows of service class s through agent a are given by

$$\mathbf{R}^a_s \mathbf{D}^a_s(\mathbf{x}^a),$$

where $\mathbf{D}^a_s(\mathbf{x}^a) := (D^a_{s,\ell}(\mathbf{x}^a) : \ell \in \mathcal{L}^a)$. In general, it is reasonable to assume that $D^a_{s,\ell}(\mathbf{x}^a)$ is a convex function of $\mathbf{x}^a$. We define

$$g_{a,s}(\mathbf{x}^a) := \max(\mathbf{R}^a_s \mathbf{D}^a_s(\mathbf{x}^a)), \quad s \in \mathcal{S},$$

to be the maximum delay experienced by service class s traffic in agent a's network.

Let $\mathbf{x} := (\mathbf{x}^a : a \in \mathcal{A})$ and $\mathcal{G} := \prod_{a \in \mathcal{A}} \mathcal{G}_a$. The problem of minimizing the total cost of all agents can be set up as the following constrained optimization problem:

$$\min_{\mathbf{x} \in \mathcal{G}} \quad \sum_{a \in \mathcal{A}} C^a(\mathbf{x}^a) \tag{2a}$$

$$\text{s.t.} \quad \sum_{a \in \mathcal{A}} R^s_{f,a} g_{a,s}(\mathbf{x}^a) \leq d^*(s), \ s \in \mathcal{S} \text{ and } f \in \mathcal{F}^s \tag{2b}$$

It is evident that the above problem in (2) is of the form in (1). Note that the decisions by the agents are coupled only via the constraints in (2b), while the objective function is separable.

III. Distributed Optimization Algorithm with Noisy Measurements

It is clear from the discussion and the example in the previous section that, in order for the agents to optimize the overall network performance with global constraints, we need a new distributed optimization framework that will allow the agents to cooperate with each other to satisfy the global constraints without incurring prohibitive overheads for information exchange. Furthermore, in order to allow different agents to use different networking technologies, equipment, and equipment vendors, it will be important to eliminate or minimize the need for the agents to exchange sensitive information that may reveal how they manage their networks.

A. Penalty Method with Noisy Measurements

One popular approach to solving a constrained optimization problem is a penalty method, which adds a penalty function to the objective function. The penalty increases with the level of violations of constraints. Although a general penalty function can be used in our problem, we assume a specific penalty function to simplify our exposition: consider the following approximated problem with a penalty function:

$$\min_{\mathbf{y} \in \mathcal{G}} \quad \sum_{i \in \mathcal{A}} \phi_i(\mathbf{y}_i) + \frac{\mu}{2N} \sum_{k \in \mathcal{K}_G} \left([g_k(\mathbf{y})]_+\right)^2 =: \Phi(\mathbf{y}), \tag{3}$$

where $\mu > 0$ is a penalty parameter, $\mathcal{K}_G$ is the set of K global constraints, ϕ_i is the (local) objective function of agent i (introduced in (1)), and $[\cdot]_+ := \max(0, \cdot)$. The original problem in (1) is recovered by letting $\mu \to \infty$. The function Φ

[1] Note that two routes used for two distinct flows may utilize the same set of links. In this case, we treat them as two separate routes.

is convex on $\mathcal{G}$ provided that ϕ_i and $g_{i,k}$, $i \in \mathcal{A}$ and $k \in \mathcal{K}_G$, are convex.

Unfortunately, the usual gradient projection method does not lead to a distributed algorithm for our problem because the penalty function and its gradient are coupled. Also, in many problems of practical interest, (a) the analytic expression of ϕ_i and its gradient are unknown to agent i and (b) only noisy measurements are available to estimate them. For instance, in the example discussed in the previous section, an agent needs to estimate its local costs from network measurements (e.g., packet loss rates), which contain measurement noise.

In order to deal with the issues of (i) noisy estimates of local costs and (ii) global constraints that couple the decision variables of agents, we propose a new distributed optimization algorithm that requires each agent i to keep an estimate $\mathbf{e}_i(t)$ of the constraint functions $\mathbf{g}(\mathbf{y}(t))$. These estimates are used to approximate the gradient of the penalty function and are updated using a consensus-type algorithm.

a) Update rule for decision variables: Consider the following iterative update rule, which is used by each agent to update its local decision variables: for every agent $i \in \mathcal{A}$ and $t \in \mathbb{N}$,

$$\mathbf{y}_i(t+1) = P_{\mathcal{G}_i}\big(\mathbf{y}_i(t) + \gamma_t \mathbf{U}_i(t)\big), \tag{4}$$

where $\mathbf{U}_i(t)$ are random vectors, and γ_t is a step size at iteration t. The initial value $\mathbf{y}_i(0)$ is chosen arbitrarily in $\mathcal{G}_i$. The random vector can be decomposed as follows:

$$\begin{aligned}\mathbf{U}_i(t) &= -\nabla\phi_i(\mathbf{y}_i(t)) - \frac{\mu}{N}\sum_{k\in\mathcal{K}_G}\nabla g_{i,k}(\mathbf{y}_i(t))[g_k(\mathbf{y}(t))]_+ \quad (5)\\ &\quad +\mu\sum_{k\in\mathcal{K}_G}\nabla g_{i,k}(\mathbf{y}_i(t))\Big(\frac{1}{N}[g_k(\mathbf{y}(t))]_+ - [e_{i,k}(t)]_+\Big) + \mathbf{V}_i(t)\\ &= -\nabla\phi_i(\mathbf{y}_i(t)) - \mu\sum_{k\in\mathcal{K}_G}\nabla g_{i,k}(\mathbf{y}_i(t))[e_{i,k}(t)]_+ + \mathbf{V}_i(t).\end{aligned}$$

The first and second terms on the right-hand side (RHS) in (5) are the gradient of the local objective function ϕ_i and the penalty function associated with global constraints, respectively. The third term models the difference between the correct gradient of the penalty function and the estimated gradient on the basis of agent i's estimates of constraint functions, namely $\mathbf{e}_i(t)$. The last term represents noise or stochastic perturbation in the estimates $\mathbf{U}(t) := (\mathbf{U}_i(t) : i \in \mathcal{A})$, the statistical distribution of which may depend on the current decision variables $\mathbf{y}(t)$. We emphasize that the exact gradients in (5) are unavailable to agent i and only a noisy estimate given by $\mathbf{U}_i(t)$ is available, which need to be computed from measurements.

b) Update rule for global constraint function estimates: In addition to updating local decision variables according to (4), each agent also updates its estimate of global constraint functions using the following update rule:

$$\mathbf{e}_i(t+1) = \sum_{j\in\mathcal{A}} w_{ij}\mathbf{e}_j(t) + \mathbf{g}_i(\mathbf{y}_i(t+1)) - \mathbf{g}_i(\mathbf{y}_i(t)), \tag{6}$$

where $W = [w_{ij} : i, j \in \mathcal{A}]$ is the weight matrix, and $\mathbf{e}_i(0) = \mathbf{g}_i(\mathbf{y}_i(0))$, $i \in \mathcal{A}$.

Clearly, the first term on the RHS of the consensus-type algorithm in (6) requires exchanging the estimates of constraint functions maintained by agents, but not their local decision variables. Furthermore, each agent $i \in \mathcal{A}$ needs to exchange its estimate $\mathbf{e}_i$ only with its neighbors specified by the weight matrix W. This update rule is designed to track the average of the constraint functions. Moreover, under the assumption that the weight matrix W is doubly-stochastic, we have

$$\begin{aligned}\mathbf{e}_{\text{avg}}(t) := \frac{1}{N}\sum_{i\in\mathcal{A}}\mathbf{e}_i(t) &= \frac{1}{N}\sum_{i\in\mathcal{A}}\mathbf{g}_i(\mathbf{y}_i(t))\\ &= \frac{1}{N}\mathbf{g}(\mathbf{y}(t)) =: \bar{\mathbf{g}}(t).\end{aligned} \tag{7}$$

As a result, the sum of the estimates maintained by all agents equals the correct value of constraint functions, and we can view $\mathbf{e}_i(t)$ as a local estimate of the average constraint functions. We note that, since agents exchange their estimates only with (direct) neighbors, the communication overheads scale linearly with the number of global constraints.

In (6), we assumed that the same weight matrix W is used for all constraints to simplify our discussion. In practice, however, agents may contribute to different sets of constraints and may not want to keep an estimate of constraint functions to which they do not contribute. In this case, we can adopt different weight matrices, one for each set of constraints with the same contributing agents.

B. Convergence Results

In this subsection, we introduce the assumptions under which agents' decision variables converge almost surely to an optimal point of (3) when they adopt the proposed algorithm given by (4) and (6).

Assumption 1: (a) The local constraint sets $\mathcal{G}_i$, $i \in \mathcal{A}$, are nonempty, compact and convex; and (b) the local objective functions ϕ_i and constraint functions $g_{i,k}$, $i \in \mathcal{A}$ and $k \in \mathcal{K}_G$, are convex and Lipschitz continuous on $\mathcal{G}_i$.

Assumption 1 is likely to hold in many, if not most, networking-related problems, such as resource allocation and performance optimization problems.

Assumption 2: The directed graph associated with the weight matrix W is strongly connected, and the weight matrix W is doubly stochastic.

This assumption simply asserts that the agents can communicate with each other over one or more hops. When this assumption is violated, it would not be possible for some agents to cooperate with each other, making it difficult to satisfy global constraints in dynamic environments.

Assumption 3: Given $\mathbf{y}(t) = \mathbf{y}$, the distribution of the perturbation $\mathbf{V}(t)$ is $\mu_{\mathbf{y}}$ and satisfies (a) $\int \mathbf{v}\ \mu_{\mathbf{y}}(d\mathbf{v}) = \mathbf{0}$ for every $\mathbf{y} \in \mathcal{G}$ and (b) $\sup_{\mathbf{y}\in\mathcal{G}} \int ||\mathbf{v}||^2 \mu_{\mathbf{y}}(d\mathbf{v}) =: \nu < \infty$.

This assumption states that, although measurements are noisy, (a) if the measurements are taken many times, the average value will be close to the correct value and (b) the noise is not large enough to prevent the agents from extracting information pertinent to the network performance.

For example, when the noise is bounded, the second part of the assumption holds, which is likely the case in practice.

Example: Before we proceed, we explain what these assumptions mean in the context of the example discussed in Section II-A. The local problems described in Section II-A are essentially a (probabilistic) routing problem. Therefore, for each ingress-egress pair (with positive traffic), the feasible local decision variables lie in an appropriate standard simplex, which is compact and convex. Assumption 2 means that the network is connected so that traffic can be routed from one agent to any other. It is reasonable to assume that network measurements are unbiased, i.e., the sample average will converge to the correct value when the number of samples is large. In this case, Assumption 3a holds. Further, in practice the network measurements are bounded and, consequently, Assumption 3b will be true.

Let us denote the optimal set, i.e., the set of optimal points of (3), and the optimal value by $\mathcal{Y}^*$ and Φ^*, respectively. The following result asserts that, under Assumptions 1 through 3, the decision variables updated in a distributed manner by the agents converge to an optimal point almost surely. Its proof can be found in [17]. Define $d_{\mathcal{Y}^*} : \mathcal{G} \to \mathbb{R}_+$, where

$$d_{\mathcal{Y}^*}(\mathbf{y}) = \inf\{||\mathbf{y} - \mathbf{y}^*||_2 \ \mid \mathbf{y}^* \in \mathcal{Y}^*\}.$$

Theorem 1: Suppose that Assumptions 1 through 3 hold. Under our proposed algorithm given by (4) and (6) with step sizes γ_t satisfying $\sum_{t\in\mathbb{N}} \gamma_t = \infty$ and $\sum_{t\in\mathbb{N}} \gamma_t^2 < \infty$, we have

$$\mathbb{P}\left[\lim_{t\to\infty} d_{\mathcal{Y}^*}(\mathbf{y}(t)) = 0\right] = 1.$$

Note that the step size condition in this theorem is standard for many iterative (stochastic) optimization algorithms and is needed to ensure the almost sure convergence under the proposed algorithm. In practice, however, a fixed step size may be used instead, and it has been shown in many cases that weak convergence hold, i.e., the optimization variables converge to a small neighborhood around the optimal set and remains there with high probability, when a sufficiently small constant step size is adopted [13].

IV. Numerical Studies

In this section, we provide numerical studies to illustrate how our proposed framework and algorithm described in Sections II and III can be used to assure E2E QoS over a telecommunication system with autonomously managed ANs and CNs. This simplified scenario highlights the key benefits of the proposed framework and some practical considerations in applying such distributed optimization methods to real world problems.

A. Description of setup

We consider one CN and two ANs - AN 1 and AN 2, as shown in Fig. 1. For the numerical study, we only consider downstream traffic that enters the CN from the Internet and is destined for either AN 1 or AN 2. The networks support two traffic classes, which require different average E2E delay guarantees. The traffic destined for AN i is given by $\mathbf{F}^i = (F_1^i, F_2^i)$, where F_s^i is the class s traffic destined for AN i.

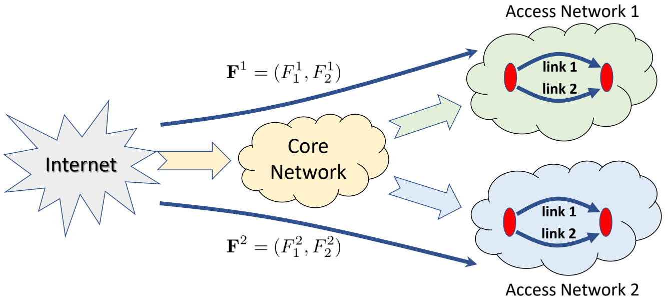

Fig. 1. Network used for numerical studies.

Each AN has two links to carry traffic (links 1 and 2 shown in Fig. 1). The local decision variables of AN i are $\mathbf{y}_i = \big((r_{\ell,1}^i, r_{\ell,2}^i, b_\ell^i) : \ell = 1, 2\big)$, where $r_{\ell,s}^i$ is the fraction of class s traffic routed through link ℓ by AN i, and b_ℓ^i is the amount of bandwidth provisioned on link ℓ by AN i. The cost of AN i comprises (i) the cost of reserving b_ℓ^i, $\ell = 1, 2$, on each link ℓ and (ii) the cost that is dependent on the average delays experienced by its traffic. More precisely, the cost of AN i, $i = 1, 2$, is given by

$$\phi_i(\mathbf{y}_i) = \sum_{\ell=1}^{2} \Big(p_\ell^i(b_\ell^i) + \sum_{s=1}^{2} \xi_s^i(r_{\ell,s}^i F_s^i, d_\ell^i(f_\ell^i, b_\ell^i))\Big), \qquad (8)$$

where $f_\ell^i = r_{\ell,1}^i F_1^i + r_{\ell,2}^i F_2^i$ is the total flow rate on link ℓ, p_ℓ^i is the function that determines the total reservation cost on link ℓ, d_ℓ^i determines the average delay on link ℓ as a function of reserved bandwidth and the total flow rate, and ξ_s^i is the delay cost function of class s traffic.

Similarly, the local decision variables of the CN are the amounts of the bandwidth provisioned for each traffic class and are given by $\mathbf{y}_c = (b_1, b_2)$, where b_s is the amount of bandwidth reserved for class s traffic. For fixed $\mathbf{y}_c$ and traffic $\mathbf{F} = (F_1, F_2)$, where $F_s = F_s^1 + F_s^2$ is the total amount of class s traffic handled by CN, its cost is given by

$$\phi_c(\mathbf{y}_c) = \textstyle\sum_{s=1}^{2} \big(p_s^c(b_s) + \xi_s^c(F_s, d_s^c(F_s, b_s))\big), \qquad (9)$$

where $p_s^c(b_s)$ is the cost of provisioning bandwidth b_s for class s traffic, d_s^c determines the delay as a function of the total rate and provisioned bandwidth for class s traffic, and ξ_s^c is the delay cost function for class s traffic.

The CN and the ANs collectively must satisfy constraints on the average E2E delays experienced by the traffic. Specifically, the average E2E delay for traffic class s should not exceed D_s. This gives rise to the following *global* constraints:

$$g_{i,s}(\mathbf{y}_i, \mathbf{y}_c) := \frac{1}{F_s^i} \sum_{\ell=1}^{2} f_{\ell,s}^i d_\ell^i(f_\ell^i, b_\ell^i) + d_s^c(F_s, b_s) - D_s \le 0,$$
$$i = 1, 2 \ \text{ and } \ s = 1, 2, \qquad (10)$$

where $f_{\ell,s}^i = r_{\ell,s}^i F_s^i$ is the amount of class s traffic traversing link ℓ.

The approximated optimization problem we are interested in solving is given by

$$\min_{\mathbf{y}\in\mathcal{G}} \Big(\big(\phi_c(\mathbf{y}_c) + \sum_{i=1}^{2}\phi_i(\mathbf{y}_i)\big) + \frac{\mu}{6}\sum_{i=1}^{2}\sum_{s=1}^{2}\big([g_{i,s}(\mathbf{y})]_+\big)^2\Big) \quad (11)$$

with decision variables $\mathbf{y} = (\mathbf{y}_c, \mathbf{y}_1, \mathbf{y}_2)$.

B. Practical Modifications of the Proposed Algorithm

Our proposed framework and convergence results in Sections II and III, respectively, guarantee the almost sure convergence of the decision variables to an optimal point when the approximated problem in (3) is convex. However, there are a few practical issues that one should consider, which are discussed here.

- **Modification 1 –** While the step sizes γ_t, $t \in \mathbb{N}$, can be chosen sufficiently small to avoid large fluctuations in the decision variables, which are updated in accordance with (4) and (6), selecting small step sizes also hampers their convergence to the optimal set. In order to skirt this issue, we apply a limiter to (some elements of) $\gamma_t\mathbf{U}_i(t)$ in (4). Specifically, we use the limiter to the elements corresponding to the routing probabilities $r^i_{\ell,s}$ so that the elements of $\gamma_t\mathbf{U}_i(t)$ corresponding to these decision variables always lie in [-0.01, 0.01]. The use of such limiters alters our proposed algorithm from that described in Section II. However, the almost sure convergence to the optimal set shown in Section III still holds with the modified algorithm under the assumed Lipschitz continuity of gradients functions and bounded noise.
- **Modification 2 –** We observe from (5) that when agent i's estimate of the kth global constraint at iteration t, namely $e_{i,k}(t)$, is positive, it introduces the corresponding term $\mu\nabla g_{i,k}(\mathbf{y}(t))[e_{i,k}(t)]_+$ in the update term used in (4). Hence, when $e_{i,k}(t)$ oscillates around zero, it is added to the update term whenever $e_{i,k}(t) > 0$ while it makes no contribution to the update term when $e_{i,k}(t) \le 0$, causing a non-negligible change in the update term. This is undesirable and slows down the convergence of decision variables.

In order to cope with this issue, rather than using the true delay budgets $D_s, s = 1,2$, we use *target* delay budgets equal to $D^T_s = \tau D_s$, where $\tau \in (0,1]$. We call the global constraints with the delay budgets replaced by the target delay budgets the *fictitious* global constraints. The main idea behind the use of fictitious global constraints is two-fold. First, satisfying the global constraints with true delay budgets in general requires selecting a large penalty parameter, which increases the sensitivity to the constraint function values (and their estimates). Second, using *fictitious* constraint functions allows us to use a smaller penalty parameter (hence, reduces the aforementioned sensitivity to the global constraint function values); we can violate the fictitious delay constraints while still satisfying the true delay constraints. This implies that the agents' estimates of fictitious constraint functions stay positive, avoiding the aforementioned oscillations, while still satisfying the original global constraints.

C. Numerical Results

In our numerical study, we assumed following functions:

$$d^i_\ell(f,b) = \upsilon_\ell\big(f/b\big)^a,\ d^c_s(f,b) = \big(f/b\big)^a, p^i_\ell(b) = \kappa^i_\ell\cdot b^k,$$
$$p^c_s(b) = \kappa_s\cdot b^k,\ \xi^c_s(f,d) = \beta_s f d,\ \text{and}\ \xi^i_s(f,d) = \beta^i_s f d,$$

where $\kappa^i_\ell > 0$, $\kappa_s > 0$, $\gamma_s > 0$, $\beta^i_s > 0$, $a \ge 1$ and $k \ge 1$. The following parameter values are used in the study:

$$\upsilon_1 = 1,\ \upsilon_2 = 0.8,\ \kappa^1_1 = 4,\ \kappa^1_2 = 5,\ \kappa^2_1 = 7,\ \kappa^2_2 = 9,$$
$$\kappa_1 = 4,\ \kappa_2 = 6, a = 2.5,\ k = 1.1,\ \beta^i_1 = \beta_1 = 40,$$
$$\text{and}\ \ \beta^i_2 = \beta_2 = 10,\ i = 1,2.$$

The flow rates are $\mathbf{F}^1 = (40,20)$ and $\mathbf{F}^2 = (50,30)$. The delay budgets are chosen to be $D_1 = 0.7$ and $D_2 = 0.5$ with the target delay budgets set to 60 percent of the true delay budgets, which are $D^T = (0.42, 0.3)$. The penalty parameter is set to $2\cdot 10^4$: a relatively large value of penalty parameter is necessary in our scenario because the delay budgets are much smaller than the costs of reserving bandwidth for the networks. The employed doubly-stochastic weight matrix $W := [w_{ij} : i,j \in \{c,1,2\}]$ used in (6) is given by

$$W = \begin{bmatrix} 3/4 & 1/8 & 1/8 \\ 1/8 & 7/8 & 0 \\ 1/8 & 0 & 7/8 \end{bmatrix}.$$

The stochastic perturbation $\mathbf{V}_i(t)$ introduced in the noisy observation of gradient is uniformly distributed over $[-\sigma\nabla\phi_i(\mathbf{y}_i(t)), \sigma\nabla\phi_i(\mathbf{y}_i(t))]$. For the reported numerical results, we pick $\sigma = 0.75$. The step sizes are selected to be $\gamma_t = \min(0.1, (t+1)^{-0.6})$, $t \in \mathbb{N}$. Finally, the decision variables are initialized as follows. Let $F^i = \sum_{s=1}^{2} F^i_s$, $i = 1,2$, and $F^T = F_1 + F_2 = F^1 + F^2$:

$$\mathbf{y}_c = (B_1, B_2),\ \mathbf{y}_1 = (0.5, 0.5, 0.5, 0.5, B^1_1, B^1_2),$$
$$\text{and}\ \ \mathbf{y}_2 = (0.5, 0.5, 0.5, 0.5, B^2_1, B^2_2),$$

where $B_s \sim \text{Uniform}(0, 2F^T)$, $s = 1,2$, and $B^i_\ell \sim \text{Uniform}(F^i, 2F^i)$, $i = 1,2$ and $\ell = 1,2$. These random variables are selected independently.

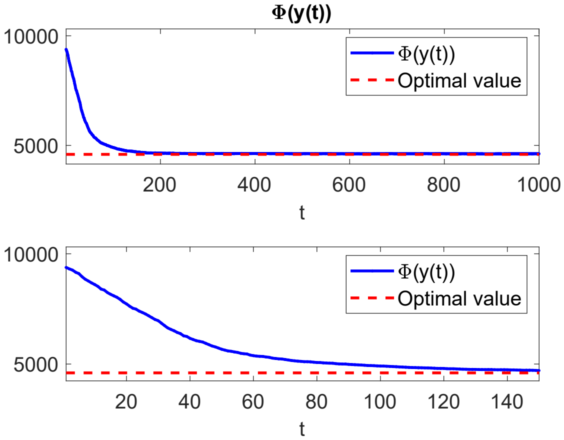

Fig. 2. Plot of the objective function $\Phi(\mathbf{y}(t))$.

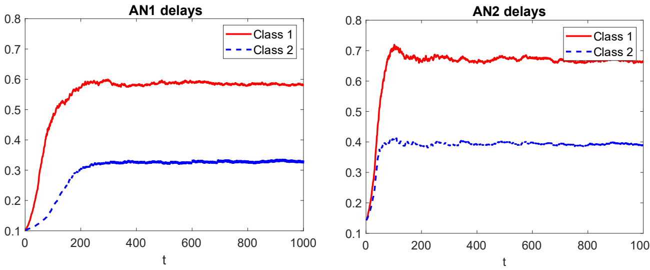

Fig. 3. End-to-end delays experienced by traffic traversing to AN 1 (left) and AN 2 (right).

First, we plot the value of the objective function $\Phi(\mathbf{y}(t))$ of the approximated optimization problem in (11) for t in $\{1, 2, \ldots, 1000\}$ (top) and $\{1, 2, \ldots, 150\}$ (bottom) in Fig. 2. The dotted red lines represent the optimal value we obtained using the MATLAB optimization tool.[2] The plots suggest that the value of $\Phi(\mathbf{y}(t))$ converges to the optimal value quickly. In fact, the bottom plot indicates that we are within 2-3 percent of the optimal value within 150 iterations, despite large noises.

Fig. 3 plots the realized E2E delays experienced by the traffic destined for AN 1 (left) and AN 2 (right). With the employed parameters, the true delay constraints are satisfied for both traffic classes. In contrast, when there is no penalty function, the realized E2E delays for class 1 and 2 traffic destined for AN 1 (resp. AN 2) are 3.7 and 8.7 (resp. 5.6 and 10.6), respectively. Obviously, these numbers are many times larger than their delay budgets and, hence, are unacceptable.

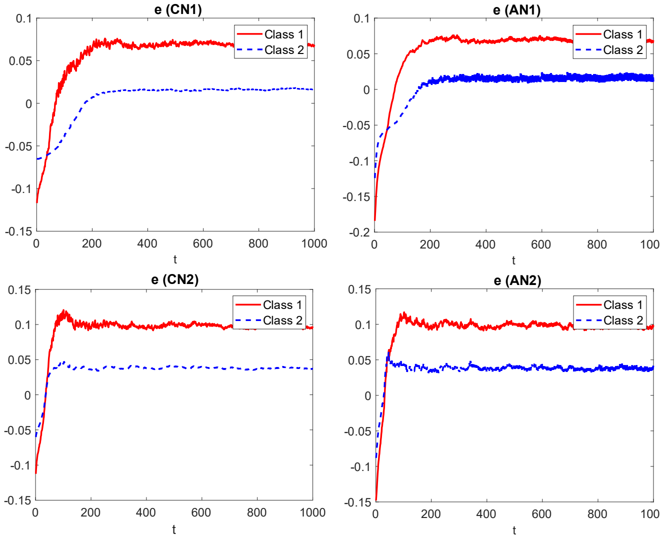

Fig. 4. Estimates of constraint functions maintained by agents.

Recall that the E2E target delay budgets for class 1 and 2 traffic are $D_1^T = 0.42$ and $D_2^T = 0.3$, respectively. The delays experienced by the traffic destined for AN 1 in the left plot of Fig. 3, which include the delays in both the CN and AN 1, exceed the target delays by approximately 0.17 and 0.03 for class 1 and 2 traffic, respectively. Hence, according to (7), the estimates of the agents must be approximately 0.06 and 0.01. Similarly, according to the right plot in Fig. 3, the agents' estimates for class 1 and 2 traffic should be approximately 0.1 and 0.03. Fig. 4 shows the estimates $\mathbf{e}_i(t)$, $i \in \{c, 1, 2\}$; the top figures (resp. bottom figures) are the estimates of $g_{1,s}$ (resp. $g_{2,s}$), $s = 1, 2$. It is clear from the figure that the agents are indeed able to closely track the average constraint function values using the proposed consensus-type algorithm in (6).

V. Discussion: Relation to Nash Equilibria of a State-Based Potential Game

In the previous sections, we implicitly assumed that the agents are willing to employ the proposed algorithm to cooperate in order to satisfy global constraints. In general, the agents will adopt the proposed algorithm only if they believe that they cannot reduce their own costs without significantly increasing global constraint violations by deviating from the equilibrium, i.e., an optimal point of (3). It turns out that there is a close relation between an optimal point of (3) and a Nash equilibrium of a state-based potential game (SBPG) associated with the problem.

An SBPG consists of the following [18]:

a) a set of agents $\mathcal{A}$;
b) a state space $\mathcal{X}$;
c) for each agent $i \in \mathcal{A}$ and each state $\mathbf{x} \in \mathcal{X}$, (i) a state-dependent admissible action set $\mathcal{Y}_i(\mathbf{x})$ and (ii) a state-dependent cost function J_i, where $J_i(\mathbf{x}, \mathbf{a})$ is agent i's cost at the state $\mathbf{x}$ when the agents adopt action profile $\mathbf{a} \in \prod_{i\in\mathcal{A}} \mathcal{Y}_i(\mathbf{x}) =: \mathcal{Y}(\mathbf{x})$; and
d) a deterministic state transition function f: given the current state $\mathbf{x}$ and agents' action profile $\mathbf{a} \in \mathcal{Y}(\mathbf{x})$, the next state is given by $f(\mathbf{x}, \mathbf{a}) \in \mathcal{X}$.

There is also a null action profile $\mathbf{a}^0$ such that $\mathbf{a}^0 \in \mathcal{Y}(\mathbf{x})$ and $\mathbf{x} = f(\mathbf{x}, \mathbf{a}^0)$ for all $\mathbf{x} \in \mathcal{X}$. In other words, the state does not change when the agents adopt the null action profile. A state-action pair $(\mathbf{x}^\star, \mathbf{a}^\star)$ is said to be a *stationary state Nash equilibrium* of the SBPG if (i) $a_i^\star \in \arg\min_{a_i \in \mathcal{Y}_i(\mathbf{x}^\star)} J_i\big(\mathbf{x}^\star, (a_i, \mathbf{a}_{-i}^\star)\big)$ for every agent $i \in \mathcal{A}$, where $\mathbf{a}_{-i}^\star$ is the action profile of all agents except for agent i, i.e., $\mathbf{a}_{-i}^\star = (a_j^\star : j \in \mathcal{A} \setminus \{i\})$, and (ii) $\mathbf{x}^\star = f(\mathbf{x}^\star, \mathbf{a}^\star)$.

Consider a game with the state given by $\mathbf{x} = (\mathbf{y}, \mathbf{e})$, where $\mathbf{y}$ is the vector of decision variables in our optimization problem, and $\mathbf{e} = (\mathbf{e}_i : i \in \mathcal{A})$ is the vector of constraint function estimates. Given a state $\mathbf{x} \in \mathcal{X}$, an action a_i of agent i is a tuple $\big(\hat{\mathbf{y}}_i, (\hat{\mathbf{e}}_{i\to j} : j \in \mathcal{N}_i)\big)$, where $\hat{\mathbf{y}}_i$ is the change in agent i's decision variables and $\hat{\mathbf{e}}_{i\to j}$ is the estimates of constraint functions agent i forwards to its neighbor $j \in \mathcal{N}_i := \{j \in \mathcal{A} \mid w_{ij} > 0\}$. Suppose that the state transitions to a new state based on the current state $\mathbf{x}$ and the chosen action profile $\mathbf{a}$ as follows:

$$\begin{aligned}
\bar{\mathbf{y}}_i &= \mathbf{y}_i + \hat{\mathbf{y}}_i \\
\bar{\mathbf{e}}_i &= \mathbf{e}_i + \mathbf{g}_i(\bar{\mathbf{y}}_i) - \mathbf{g}_i(\mathbf{y}_i) + \textstyle\sum_{j\in\mathcal{N}_i} \big(\hat{\mathbf{e}}_{j\to i} - \hat{\mathbf{e}}_{i\to j}\big).
\end{aligned}$$

[2] Mention of commercial products in this paper is for information only; it does not imply recommendation or endorsement by NIST.

Consider the cost function of agent i given by

$$J_i(\mathbf{x}, \mathbf{a}) = \phi_i(\bar{\mathbf{y}}_i) + \frac{\mu}{2} \sum_{j \in \mathcal{N}_i} \|[\bar{\mathbf{e}}_j]_+\|_2^2, \tag{12}$$

where $\mu > 0$ is a trade-off parameter. This gives rise to an SBPG with a global potential function given by [16]

$$\Phi_\mu(\mathbf{x}, \mathbf{a}) = \sum_{i \in \mathcal{A}} \phi_i(\bar{\mathbf{y}}_i) + \tfrac{\mu}{2} \sum_{i \in \mathcal{A}} \|[\bar{\mathbf{e}}_i]_+\|_2^2. \tag{13}$$

Note the similarity between the objective function in (3) and the global potential function in (13). From (7), we have $\mathbf{g}(\mathbf{y}(t)) = \sum_{i \in \mathcal{A}} \mathbf{g}_i(\mathbf{y}_i(t)) = \sum_{i \in \mathcal{A}} \mathbf{e}_i(t)$. Hence, when all agents have the correct estimate of the average constraint function values, i.e., $\mathbf{e}_i(t) = \mathbf{e}_{\text{avg}}(t)$ for all $i \in \mathcal{A}$, the global potential function coincides with the objective function in (3) with a different penalty parameter. This observation can be used to prove the following proposition.

Proposition 1: Suppose that $\mathbf{y}^* \in \mathcal{Y}^*$ is an optimal point of (3). Then, $\big((\mathbf{y}^*, \mathbf{g}(\mathbf{y}^*)/N), \mathbf{0}\big)$, where $\mathbf{0}$ is the null action profile, is a stationary state Nash equilibrium of the SBPG.

The above proposition means that each optimal point in $\mathcal{Y}^*$ leads to a stationary state Nash equilibrium and there is an one-to-one mapping from $\mathcal{Y}^*$ to a set of stationary state Nash equilibria of the SBPG.

VI. Conclusions

We proposed a novel distributed algorithm that enables autonomously managed ANs and CNs in a telecommunication system to cooperate with each other to meet E2E KPI goals. The new algorithm overcomes two major limitations of prior techniques. First, unlike prior approaches that require static local KPI budgets for each autonomous subsystem, the new algorithm allows the autonomous subsystems to dynamically negotiate their local KPI budgets. Second, rather than requiring the autonomous subsystems to share their local decision variables with each other, our proposed algorithm only needs the subsystems to exchange their estimates of the global constraint functions. We proved that the new algorithm converges to an optimal solution almost surely, and presented numerical results to demonstrate that the convergence occurs quickly even with measurement noise.

References

[1] 3GPP TS 22.261 V15.8.0 (2019-09), "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service requirements for the 5G system; Stage 1 (Release 15)."

[2] 3GPP TS 23.501 V15.12.0 (2020-12), "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; System architecture for the 5G System (5GS); Stage 2 (Release 15)."

[3] 3GPP TS 28.550 V16.1.0 "Management and orchestration; Performance assurance".

[4] 3GPP TS 28.552 V16.1.0 "Management and orchestration; 5G performance measurements".

[5] 3GPP TS 28.554 V16.0.0 "Management and orchestration; 5G end to end Key Performance Indicators (KPI)".

[6] 3GPP TS 32.425 V16.3.0 "Performance Management (PM); Performance measurements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN)".

[7] 3GPP TS 28.532 V16.0.0 "Management and orchestration; Generic management services".

[8] M. Asad, A. Basit, S. Qaisar and M. Ali, "Beyond 5G: Hybrid End-to-End Quality of Service Provisioning in Heterogeneous IoT Networks," *IEEE Access*, 8:192320-192338, 2020, doi: 10.1109/ACCESS.2020.3032704.

[9] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Trans. on Automatic Control*, 58(2):391-405,

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[11] D. L. C. Dutra, M. Bagaa, T. Taleb and K. Samdanis, "Ensuring End-to-End QoS Based on Multi-Paths Routing Using SDN Technology," Proc. of IEEE GLOBECOM, Singapore, 2017, pp. 1-6, doi: 10.1109/GLOCOM.2017.8254076.

[12] R. Jacquet, G. Texier and A. Blanc, "Computing end-to-end QoS paths in the Internet considering multiple alliances," Proc. of the 16th International Telecommunications Network Strategy and Planning Symposium (Networks), 2014, pp. 1-6, doi: 10.1109/NETWKS.2014.6959248.

[13] H. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. Springer, 2003.

[14] Y. Lee, K. John and R. Vilalta, "Extended ACTN Architecture to Enable End-To-End 5G Transport Service Assurance," Proc. of the 21st International Conference on Transparent Optical Networks (ICTON), Angers, France, 2019, pp. 1-3, doi: 10.1109/ICTON.2019.8840270.

[15] K. B. Letaief, W. Chen, Y. Shi, J. Zhang and Y. A. Zhang, "The Roadmap to 6G: AI Empowered Wireless Networks," *IEEE Communications Magazine*, 57(8):84-90, Aug. 2019, doi: 10.1109/MCOM.2019.1900271.

[16] N. Li and J.R. Marden, "Decoupling coupled constraints through utility design," *IEEE Trans. on Automatic Control*, 59(8):2289-=2294, Aug. 2014.

[17] V. S. Mai, R. J. La, T. Zhang and A. Battou, "Distributed Optimization with Global Constraints Using Noisy Measurements," arXiv preprint arXiv:2106.07703, 2021.

[18] J. Marden and J. S. Shamma, "Game theory and distributed control," *Handbook of Game Theory with Economic Applications*, pp. 861-899, 2015.

[19] W. Saad, M. Bennis and M. Chen, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems," *IEEE Network*, vol. 34(3):134-142, May/June 2020, doi: 10.1109/MNET.001.1900287.

[20] C. She et al., "Deep Learning for Ultra-Reliable and Low-Latency Communications in 6G Networks," *IEEE Network*, 34(5):219-225, Sep./Oct. 2020, doi: 10.1109/MNET.011.1900630.

[21] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," *IEEE Journal of Selected Topics in Signal Processing*, 5(4:)772-790, 2011.

[22] C.-C. Teng, M.-C. Chen, M.-H. Hung and H.-J. Chen, "End-to-end Service Assurance in 5G Crosshaul Networks," Proc. of the 21st Asia-Pacific Network Operations and Management Symposium (APNOMS), Daegu, Korea (South), 2020, pp. 306-309, doi: 10.23919/APNOMS50412.2020.9236977.

[23] Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen and X. Li, "End-to-End Quality of Service in 5G Networks: Examining the Effectiveness of a Network Slicing Framework," *IEEE Vehicular Technology Magazine*, 13(2):65-74, Jun. 2018, doi: 10.1109/MVT.2018.2809473.

[24] T. Zhang, "Toward Automated Vehicle Teleoperation: Vision, Opportunities, and Challenges," *IEEE Internet of Things Journal*, 7(12):11347-11354, Dec. 2020, doi: 10.1109/JIOT.2020.3028766.

[25] G. Zhu, J. Zan, Y. Yang and X. Qi, "A Supervised Learning Based QoS Assurance Architecture for 5G Networks," *IEEE Access*, 7:43598-43606, 2019, doi: 10.1109/ACCESS.2019.2907142.