Slice as a Service (SlaaS) Optimal IoT Slice Resources Orchestration

Vincenzo Sciancalepore, Flavio Cirillo, Xavier Costa-Perez
NEC Laboratories Europe, Germany

***Abstract*—The increasing deployment of smart devices using mobile networks is pushing operators to consider efficient ways to tailor their infrastructure to the Internet of Things (IoT) diverse requirements and traffic characteristics. A promising approach to address this need is the novel concept of *network slicing*, which aims at allocating portions of network resources to specific tenants, such as enhanced mobile broadband (eMBB), IoT, e-health, connected vehicles, etc. While this has been traditionally done with long-term agreements between network operators and tenants as MVNOs, in this work we focus on a new business model where network operators offer network slices as a service (SlaaS). In particular, we propose a novel system comprising an *IoT Broker* managing massive IoT network slices services and a *Network Slice Broker* that through bi-directional negotiations are able to efficiently allocate and orchestrate network resources.**

I. Introduction

With the continuously increasing number of connected Internet-of-Things (IoT) devices, foreseen as tenths of Billion by 2020, as well as with even more IoT use-case enablers, the IoT world has got its momentum in the small and medium-sized enterprises (SMEs) market. To this aim, the evolution of new network technologies is enlarging its horizon by exhibiting as one of their strengths a bigger flexibility on network definition and network virtualization. The context of the 5th generation of mobile network, namely 5G, is further enriched when the *network slicing* concept comes into play.

The Next Generation Mobile Network Alliance (NGMN) defines the 5G network slice as a driver paradigm to run multiple self-contained logical networks as independent business operations on a shared physical infrastructure [1]. Therefore, each network slice represents a virtualized independent end-to-end network allowing infrastructure providers to deploy different architectures in parallel. Leveraging on this concept, the infrastructure providers may customize their own networks by opening their facilities to novel business players, such as virtual mobile network operators (VMNOs), third-parties as well as Over-The-Top (OTT) applications, as shown in Fig. 1. Such entities behave as tenants of the same physical infrastructure. This brings new challenges in designing a network resources allocation policy, which must guarantee the resource isolation principle and, at the same time, it improves the multiplexing gain resulting in a more cost-effective resource allocation for the infrastructure provider.

While the flexibility introduced with network slicing dynamics fosters a network virtualization evolution, infrastructure providers do not quantify yet the real benefit brought to their current business cases. There is a real need of assessing and

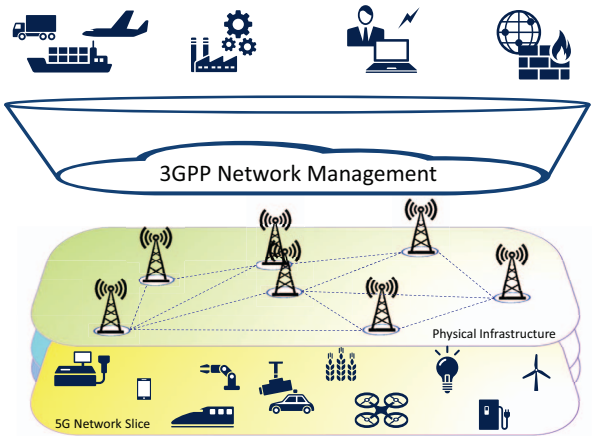


Fig. 1: 5G Network Slicing concept

brokering the network slicing operations between infrastructure providers and different tenants. Recently, a novel logically centralized entity was defined, namely a *capacity broker* [2], considering its mapping into current 3GPP architectures and in charge of network slicing admission control operations. This functional block has been extensively improved in [3] to provide a means for optimally allocating and configuring Radio Access Network (RAN) slices based on on-demand network slice requests. Therefore, the network operators efficiently face with the network paradigm change by providing network slicing capabilities as a service, namely Slice as a Service (SlaaS).

The main benefit introduced by a multi-tenant-enabled network is the ability for heterogeneous industrial segments to acquire and use the same network infrastructure. The IoT world can be envisioned as the most suitable customer exploiting a self-managed and isolated slice of network resources, given the high heterogeneity level of its traffic requirements [4]. The advantage of engaging IoT traffic is two-fold: (i) it is flexible enough to be reshaped based on the network conditions, (ii) it may require advanced Service Level Agreements (SLAs) for very short periods, which, in turn, translate into an additional profitable gain for the network operator. Fig. 2 introduces an IoT system built on top of a 5G network slice, wherein sensors communicate wirelessly (e.g., Bluetooth, ZigBee, LoRa) with an IoT Gateway (GW), which is capable of handling, storing and exposing data through 5G facilities. The IoT platform, then, enables the communications between IoT applications and "things" in a service-oriented fashion.

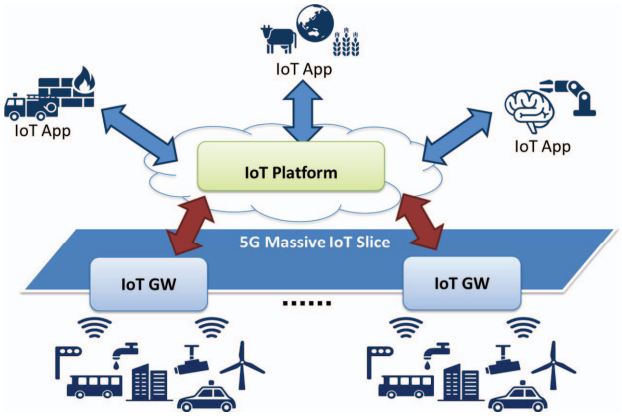


Fig. 2: IoT System deployed on 5G Network Slice

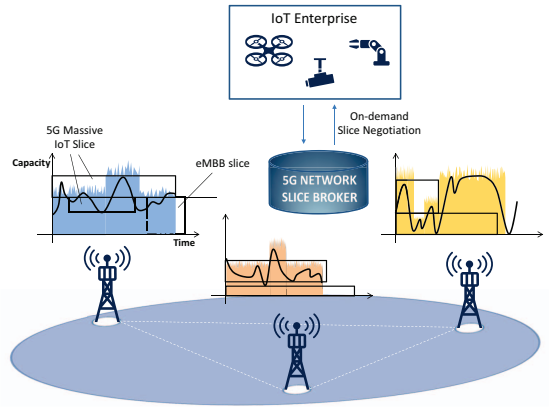


Fig. 3: 5G Network Slicing Brokering operations

A number of research works proposed to optimize IoT traffic flows based on different Quality of Service (QoS). An example is provided by [5], where an advanced resource allocation scheme is suggested to optimally select IoT data streams to be processed in a cloud environment and to drastically reduce the upload bandwidth by means of prediction algorithms. Therefore, the idea of an *IoT Broker* in charge of interacting with external application while optimally delivering different data information is taking off, as suggested in [6] where the authors propose a shortest processing time algorithm for scheduling web-based IoT messages. In addition, a central IoT Broker is best suited to shape IoT traffic using service-aware QoS at application level. Some research has been done in order to analyze QoS requirements for the IoT domain [7] including specific IoT services. [8] goes a step forward and proposes an optimization approach for a specific class of IoT traffic, such as periodic reports of sensor networks, within given network resources. This scheme permits to minimize the size of the network capacity without incurring in traffic congestions for occasional bursts of traffic.

None of those solutions considers an IoT Broker with a slicing traffic-aware feature. To the best of our knowledge we pioneer the idea of blending together the IoT traffic reshaping features with a 5G network slice broker, pursuing the network utilization maximization and QoS satisfaction. Differently from well-known game theory approaches, coalition schemes and sealed-bid systems with resource efficiency or profit objectives ([9], [10]), the work in this paper targets solving the provider-customer problem of efficiently allocating/maintaining/configuring 5G network slices for IoT tenants.

In particular, this work provides the message flow between the IoT platform and the 5G network management responsible for properly configuring network slices to (i) limit (e.g., by changing granularity, quality and frequency [11] of requested information) the IoT messages load when 5G network congestions occur, (ii) rescheduling IoT messages for underloaded periods of time, (iii) prepare 5G network facilities (e.g., by rescaling other network slices, offloading, denying) when mission-critical messages must be exchanged between IoT players in safety contexts.

II. Problem Statement

Along the direct communication between infrastructure provider and tenant, several challenges are identified. On the one hand, the infrastructure provider applies and defines optimization algorithms for maximizing the allocation of virtual network slices aiming at the maximization of the revenues. On the other hand, the tenants (network slice customers) aim at minimizing the slice parameters with the dual target of minimizing the costs while keeping affordable the Quality of Service required. This usually results in a sub-optimal slicing configuration and limited resource monetization.

In the following, we provide a detailed analysis of IoT traffic reshaping features and 5G network slice capabilities, separately. This paves the road towards our optimized joint solution, presented in Section III.

A. IoT Broker

IoT communications are enabled by means of an *IoT Broker component* [12]. This entity serves as a middleware that exposes to IoT applications a northbound interface for IoT data requests, which can be conveyed either through a conventional network means, which does not involve resources limitation issues (e.g., fixed-line technology means), or through massive IoT network slices, as shown in Fig. 2. The IoT application developers seamlessly interact with the IoT Broker that takes care of interfacing with different IoT Gateways (GWs) to retrieve all needed information.

The interaction between IoT applications and IoT GWs can be synchronous and asynchronous. The former is a query-response interaction, where applications inquire for data, and expect a single response. The latter follows the subscription-notify paradigm: once subscribed to an IoT data service, the IoT application is notified with the corresponding data, when an update occurs. In general, the IoT Broker maps the northbound requests onto a set of southbound requests to IoT GWs. Conveniently, the subscription-notify paradigm activates data flows of an IoT service between IoT platform and IoT GWs upon the reception of the subscription request, differently from a publish-subscribe scheme where data is continuously pushed to the IoT platform regardless of the IoT applications'

interests. The IoT Broker is meant here to be a single logical component, which can be scaled if necessary. How to reach this scalability is out of the scope of this work.

B. 5G Network Slice Broker

The *5G Network Slice Broker* is the network component in charge of (i) interfacing with external network tenants in order to accept/reject new network slice requests, (ii) managing the slice instantiation/resize/maintenance/deletion operations, (iii) monitoring slice traffic [13]. Fig. 3 shows a self-explained example, where the 5G Network Slice Broker receives a new slice request from an IoT Enterprise asking for a certain amount of network resources. If network capabilities are enough to accommodate the new slice request, the 5G Network Slice Broker instantiates a new slice by instructing the RAN elements to dedicate a given portion of resources. This example mostly focuses on the network slicing management on RAN premises. However, it can be readily extended to the transport and core network elements.

Decoupled optimizations applied on both concepts drive the system toward suboptimal states, as no direct installed communication prevents the system from reacting dynamically.

III. Joint IoT Slice Traffic-aware Solution

When the 5G network slice customer-provider relationship is broken down, the network slice brokering process acquires more knowledge about the real tenant application needs thereby optimally instantiating/configuring/scaling network slicing resource requests. Conversely, the IoT traffic might also be optimally reshaped to account for unexpected 5G network congestions.

Fig. 4 depicts a system architecture where the IoT Broker and 5G Network Slice management communicate and efficiently orchestrate IoT traffic and/or network slicing operations. IoT-related messages reach the IoT platform (data-plane) or the IoT GWs (control-plane) through the IoT massive 5G Slice. Control-plane messages are meant to dynamically adjust the amount of data exchanged between the IoT GWs and the IoT Broker. The IoT Broker might decide to enhance the quality of the data traffic (e.g., more fine-grained, smaller sampling period) when 5G network premises are underutilized or, conversely, it might ask to worsen the data quality (or delay after collecting a larger set of them), when network condition degradation or network resources preemption occur. Network resources assigned to the 5G massive IoT slice are dynamically managed by the 5G Network Slice Broker through a dedicated control-plane channel. This communication might trigger a slice resources update to cope with unexpected and rapid network changes (such as network congestions, additional network slice instantiations, IoT SLAs re-negotiations).

Interestingly, the IoT Broker provides different features: (i) shaping the IoT traffic (transmitted through the IoT massive 5G slice) by properly choosing the set of IoT GWs for satisfying the data request, or the QoS parameters of the southbound subscriptions (e.g., data granularity, notification frequency), (ii) measuring the data traffic in order to monitor

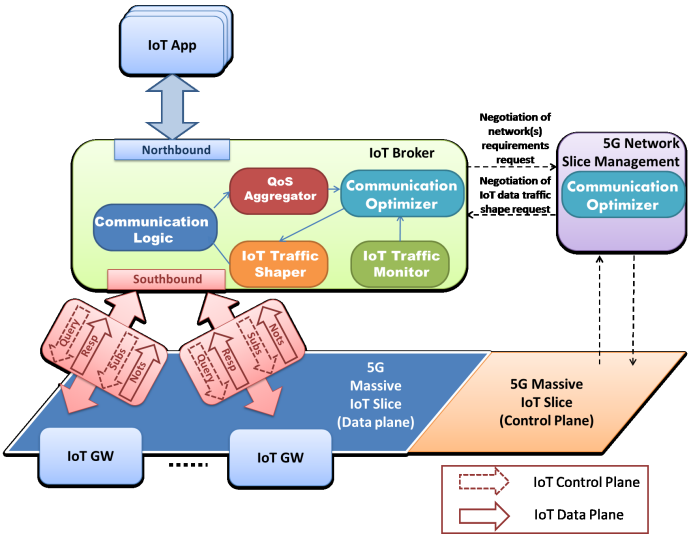


Fig. 4: Joint IoT/Slice Broker Orchestration Architecture

traffic fluctuations, changes of notification frequency (from the IoT GWs), changes of QoS parameters in query/subscriptions messages, (iii) optimizing the data traffic in order to maximize the utilization efficiency of the 5G network slice resources while still satisfying IoT applications QoS requirements.

A. Interface and Data exchange

A functional view of our system solution is shown in Fig. 4. When requests arrive from the application layer, minimum QoS requirements are evaluated against the status of the IoT traffic load. The *QoS Aggregator* functional block aggregates overlapping QoS requirements coming from different applications. For example, road traffic condition of the same area with a certain granularity might be requested for traffic light coordination and for bus scheduling optimization. The output of the *QoS Aggregator* is a unique set of QoS requirements per area by considering the more stringent ones. In addition, QoS models all the synchronous requests into aggregated minimum QoS requirements for each service, similarly to asynchronous traffic. These new requirements are then further aggregated to the QoS requirements of the asynchronous communication.

Aggregated QoS parameters are fed to the *Communication Optimizer* (CO) functional block, which finds a good trade-off between minimizing the SLA of the 5G massive IoT slice and satisfying the QoS requested. In case of IoT slice resource over-provisioning, it might trigger the 5G Network Slice Broker to *dynamically re-negotiate the slice size*. As output, the CO issues the actual QoS parameters (quality of data, frequency of data sampling and granularity of data aggregation) for each service.

The *IoT Traffic Shaper* (TS) functional block is in charge of shaping the southbound traffic, given as input the QoS parameters. TS chooses the IoT GWs, providing requested services (e.g., crowd behaviour, video surveillance, traffic situation) within a given geographical scope, and, for each of them decides specific QoS parameters carried on the southbound request. Then the *Communication Logic* (CL) functional block

(i) binds the northbound request to the southbound requests, (ii) issues the southbound requests to the IoT GWs and, (iii) once the IoT data is flowing from the southbound interface, it returns messages to the IoT applications.

The southbound IoT traffic is continuously monitored by the *IoT Traffic Monitor* (TM) functional block, which feeds back CO with the current status of the IoT network slice. The traffic load might increase or decrease due, for example, to sensors availability or changes on the data produced for critical events (e.g., a big crowded event).

When current network slice resources are not enough to accommodate reshaped IoT data traffic requests without infringing QoS requirements, CO might request the 5G Slice Broker to scale up the network slice size. The 5G Network Slice Broker is constantly monitoring the traffic of instantiated 5G network slices by checking that the traffic does not exceed the network slice boundaries. When a re-negotiation is required, the 5G Network Slice Broker checks internally whether it is feasible to satisfy such a request without breaking other slice SLAs. If the network conditions allow the 5G Network Broker to accommodate this network slice resources upgrade, the 5G Network Slice Broker provides the new amount of granted resources to the CO.

Similarly, the 5G Network Slice Broker may detect an over-provisioning of available 5G resources in one of already running network slices and, it might request CO to reshape the IoT data traffic. Automatically, the *Communication Optimizer* calculates a feasible solution for the IoT resources allocation. If a successful solution is found, the IoT Broker responds back to the 5G Network Slice Broker with a scaled amount of network slice resources.

To further validate our architecture, in the next sections we show in details two use-cases wherein this joint orchestration is strongly recommended. We then propose a problem, mathematical analysis and practical algorithm for one of those use-cases, in Section IV.

B. Use case: City Council Network Slice

We envision a city council owning a virtualized infrastructure (or network slice prior granted) and offering (part of) it as a slice service to different municipal domains, e.g., police, homeland security, public transportation companies, domestic energy providers, and markets association (shops or shopping malls). Both IoT Broker and Network Slice Broker reside within the same administrative premises pursuing the same objective: maximization of network utilization and reduction of QoS violations. In this case, IoT Broker, as an infrastructure tenant, manages the massive IoT network slice. Other tenants can be envisaged as the other municipal domains sharing the same physical infrastructure.

Different IoT services might have different QoS parameter values (see Table I). For example, *Road Situation* service might be implemented as a set of compound sensors (comprehensive of road occupancy sensor, car counting and speed sensor together with an embedded algorithm for computing a normalized value of traffic situation) installed on the streets.

TABLE I: Examples of QoS parameters for different IoT Services

	Data Quality	Geographical Granularity	Notifications Frequency
Video Surveillance	1.3, 3, 5, 8, 10 Megapixels	Single Camera, Grid 8x8 for: Building, Streets, Neighbourhood, City	10 fps, 1 fps, 0.5 fps, 0.1 fps, 0.05 fps, 0.015 fps ($\sim$ 1 per minute)
Road Situation	Historical Record, Averaged last 10m, Only Car Count & Speed Average, Norm. Traffic Situation	Sensors, Averaged by: Building, Streets, Neighbourhood, City	10s, 30s, 1m, 5m, 10m, 30m, 1 hour
Air Quality	All Sensed Particles, Pollutant levels, Norm. Air Quality	Sensors, Averaged by: Building, Streets, Neighbourhood, City	10s, 30s, 1m, 5m, 10m, 30m, 1 hour
Crowd Behaviour	WiFi Sniffer Record, Devices Detected, Crowd Pattern, Crowd Estimation	Sensors, Averaged by: Building, Streets, Neighbourhood, City	30s, 1m, 5m, 10m, 30m, 1 hour

A low data quality level could be the retrieval of only a summarized value of the traffic situation whilst a higher data quality could allow the transmission of all the aggregated sensed information (i.e., car counting, speed average, road occupancy in the last hour) till the highest quality level, which configures the transmission of all recorded data, such the speeds history of cars using the road. An orthogonal QoS parameter is the geographic granularity data. Using again the example of the Road Situation service, the data transmitted over the network slice might be with the highest granularity (retrieving all real compound sensors data) or with a lower granularity (assuming a data aggregation performed by the IoT GW per geographical scope, such as urban blocks, streets or neighborhoods). Also the timing of the messages sent through the network slice can be controlled. In this case, the IoT GW might send messages every 10s, 30s, 1 min, 2 min, 5 min, 10 min, 30 min or 1 hours.

As another example of IoT services, we might account for the *Video Surveillance*. The quality of the data is the resolution of the recorded images. Changing the granularity automatically configures the IoT GWs to send a grid of frames coming from different cameras surrounding the same geographical object, with a total amount of pixel indicated by the quality of data. In case the number of cameras is larger than the allowed number of cells of the grid, the IoT GW chooses images to be sent through certain selection algorithms, not analyzed here. The frequency of the messages affect the bitrate of the data flow, such as 30 fps, 10 fps, 1 fps, 0.5 fps, 0.1 fps, 0.05 fps, 0.015 fps ($\sim$ 1 per minute). In Table I, we have summarized QoS configurations also for: *Air Quality* service, in charge of monitoring the pollution of the air and based on weather compound sensors; *Crowd Estimation and Behaviour* service, which infers the crowd estimation in public spaces and their mobility pattern based on Wi-Fi packages sniffers.

C. Use-case: Cooperation between different domains

We also target a different network use-case: network operator owning the Network Slice Broker [14], whereas a massive IoT network slice is already instantiated (with an IoT Broker). 5G network Slice Broker and the IoT Broker belong to different administrative domains so that they selfishly aims to increase own network performance. An interaction may be established aiming at: improving the resource utilization efficiency (from the network operator perspective); reducing the network slice cost (from the IoT tenants perspective) by offering a better utilization fee for a flexible and dynamic adaptable slice. Pricing models used for such a relationship are out of the scope of this work.

While those user scenarios lay the basis for this novel cooperation between 5G Network slice management and IoT world, they are not intended to be exclusive or exhaustive. However, they provide a solid basis for evaluating and fostering the adoption of such jointly orchestration. In the next section, we present an insightful analysis of such a problem shedding the light on practical algorithmic solutions.

IV. Analysis and Practical Solution

We rely on the city council network slice use case, wherein the compound effect of an IoT Broker and a 5G network slice management benefits the same administrative domain, as explained in Section III-B.

The IoT application subscribes a particular service a asking for sensor information updates within a given geographical scope s. We assume that our system area $\mathcal{S}$ is a grid divided into multiple areas $s \in \mathcal{S}$ not overlapping [1]. Services a might be, and not limited to, the examples listed in Table I. Each of these services results in a certain amount of data delivered by the IoT Broker through the 5G network facilities. The subscription request will be issued for any single area s and is defined as $r_s = (a_s, \gamma_s, \omega_s, \mu_s, \rho_s)$. $\gamma_s \in \Gamma$ index specifies the granularity of such information, such as individual sensor, building, street, neighbourhood or urban context. Moving from fine-grained granularity (sensors level) to data aggregation may require IoT GWs to perform data analytics operations by considering average, peak or other statistical values. While this drastically reduces network slice congestions, it may prevent the IoT application from acquiring detailed information. Furthermore, we also denote the frequency of data updates with index $\omega_s \in \mathcal{W}$, which may range from 10s to 5 min. $\mu_s \in \mathcal{Q}$ specifies the quality of delivered information, whereas $\rho_s \in \mathcal{P}$ defines the priority of the subscription request. Please note that a higher priority index corresponds to a higher cost for being guaranteed with required parameters [2]. All these sets are discrete sets of values that drive the overall system utilization, as diverse service configurations require different amounts of data exchanged. Our joint system leverages on this powerful trade-off to accommodate first high-priority service requests into the available network slice capacity while differing low-priority traffic flows.

Service requests for different geographical areas come periodically and are processed by the IoT Broker. As shown in Fig. 4, the IoT Broker forwards such request parameters to a *Communication Optimizer* functional block, in charge of optimally scheduling service requests into the slice capacity following a priority-based policy. *Communication Optimizer* block is defined as a dual functional entity: it aims to properly reshape the service configuration in order to fulfil the network slice capacity limits. As soon as the application service requests exceed the network slice capacity, it may trigger its counter-part on the 5G network management to promptly adjust the network slice boundaries if no network congestions occur. Analytically, we can express the optimization problem as follows

Problem `IoT-Optimizer`:

$$\begin{aligned} \text{maximize} \quad & \sum_{s\in\mathcal{S}} (x_s - p_s) \\ \text{subject to} \quad & \sum_{s\in\mathcal{S}} g_a(\gamma_s) q_a(\mu_s) f_a(\omega_s)(x_s - p_s) \leq \eta_{\text{o}}; \\ & x_s \geq \quad _{\{\rho_s \geq \bar{\rho}\}}, \quad \forall s \in \mathcal{S}; \\ & x_s, p_s \in \{0;1\}, \quad \forall s \in \mathcal{S}; \end{aligned}$$

where $f_a(\cdot), g_a(\cdot), q_a(\cdot)$ are discrete functions providing the amount of datarate given certain frequency, granularity and quality values, respectively, for a particular service a, whereas x_s is a binary value indicating whether the service request for area s can be scheduled. The datarate might change over time due to a new IoT resources availability (e.g., new sensors installed under an IoT service) or because of critical situations (e.g., more data is generated outside a stadium right after a crowded event). The *IoT Traffic Monitor* measures the IoT traffic and updates the $f_a(\cdot), g_a(\cdot), q_a(\cdot)$ functions. η_{o} is the capacity assigned to the IoT network slice and p_s is a binary value (penalty) indicating that the service request for area s cannot be admitted within the current amount of assigned network slice resources. $\bar{\rho}$ is a threshold defined by the *Communication Optimizer* to force the IoT Broker to accept service requests above a certain priority level (e.g., security issues). The main constraint defines the total amount of data exchanged $(g_a(\gamma_s) q_a(\mu_s) f_a(\omega_s))$ per area s by the IoT devices, which must be scheduled into the network slice traffic capacity. The tuple $(\gamma_s, \mu_s, \omega_s)$ is sent to the *IoT Traffic Shaper*, which adjusts the service configurations on the IoT GWs, accordingly. When $\sum_{s\in\mathcal{S}} p_s \geq 0$, some IoT service requests cannot be accommodated and an intervention is required by the network slice management. Therefore, the *Communication Optimizer* sends an upgraded network slice request, which comprises $\lambda_{\text{o}} = \eta_{\text{o}} + p_{\text{o}}$, where $p_{\text{o}} = \sum_{s\in\mathcal{S}} p_s \left(g_a(\gamma_s) q_a(\mu_s) f_a(\omega_s)\right)$ is the exceeding capacity required by the IoT slice. On the other side, the *Communication Optimizer* installed on the 5G network slice management deals with the following optimization

[1] This assumption makes tractable the problem analysis. However, it can be easily extended for advanced cases with overlapping spatial areas.

[2] We assume that the IoT entity aims at minimizing the overall cost while getting an acceptable level of quality. A detailed discussion about this mechanisms is out of the scope of this dissertation. However, advanced mechanisms for optimally adjusting such priority/cost levels might be considered without affecting the problem analysis presented in this paper.

problem to optimally scheduled its network resources between different network services $i \in \mathcal{I}$

Problem `Network-Optimizer`:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in \mathbb{I}} \eta_i \\ \text{subject to} \quad & \sum_{i \in \mathbb{I}} \eta_i \leq C_{\text{slice}}; \\ & \eta_i \geq \lambda_i, \quad \forall i \in \mathcal{I}; \\ & \eta_i \in \mathbb{R}_+, \quad \forall i \in \mathcal{I}; \end{aligned}$$

where λ_i is the amount of network resources in terms of datarate required by the tenant i, whereas C_{slice} is the slice capacity managed by the 5G controller[3]. The output directly feeds back the IoT Broker (through the *Communication Optimizer* block) by specifying the updated amount of resources assigned to the IoT slice (i.e., $\eta_{i=\text{o}}$).

Interestingly, Problem `IoT-Optimizer` and Problem `Network-Optimizer` are tightly connected, as the output of one provides the input for the other one and vice-versa. This implies that in case of emergency or security threats, if the slice size is not enough (η_{o}), the IoT Broker may trigger an update request to the 5G network management (λ_{o}) and promptly get augmented network resources to accommodate the traffic burst. Conversely, when the 5G network experiences congestions, it might ask the IoT Broker to reduce the resources utilization (η_{o}) so as to make room for other network services i.

A. Algorithm Description

Problem `IoT-Optimizer` above-described is an Integer Linea Programming (ILP) problem. Such class of problems is known to be NP-Hard [15]. For the sake of brevity, we skip the formal proof of NP-Hardness and NP-Completeness. However, small instances of the problem with few levels of granularity, frequency and quality may help in finding an optimal solution within a polynomial time by means of an exhaustive search.

Here we present a heuristic solution for solving Problem `IoT-Optimizer`, as most of the ready-to-use algorithms in the literature properly address Problem `Network-Optimizer`. The pseudocode is listed in Algorithm 1. The IoT Slice-aware Traffic Optimizer splits the service requests into two subsets: in the former there are only high-priority service requests, i.e., where $\rho_s \geq \bar{\rho}$ whereas in the latter all the other services requests s. After placing high-priority requests within the slice capacity η_{o} following a decreasing order (the rationale is based on first fit decreasing schemes proposed in the literature for well-known knapsack problems), our algorithm tries to place as many low-priority requests as possible that still fit into the available slice capacity. If some high-priority service requests are not scheduled yet, the algorithm increases the capacity limit at expenses of a penalty value p_s. In this case, the system realizes that the current slice configuration is not enough to satisfy high-priority instances requirements, and it automatically triggers the 5G network management for updating the slice capacity boundaries.

[3]We assume that the slice capacity is properly designed from the network management perspective. If network congestions occur, Problem `Network-Optimizer` might be unfeasible so that the 5G network slice management needs to discard some network slice request updates, based on a priority basis. For further information, we refer the reader to advanced 5G network slicing brokering mechanisms as explained in [3].

Algorithm 1 IoT Slice-aware Traffic Optimizer

1) Initialise sets $\mathcal{K} \leftarrow 0$ and $\mathcal{J} \leftarrow 0$.
2) Update high-priority set $\mathcal{K} \leftarrow s : \rho_s \geq \bar{\rho}$ and low-priority set $\mathcal{J} \leftarrow s : \rho_s < \bar{\rho}, \forall s \in \mathcal{S}$.
3) Sort $\mathcal{K}$ in a decreasing order and $\mathcal{J}$ in an increasing order based on $g_a(\gamma_s)q_a(\mu_s)f_a(\omega_s)$.
4) Place $s \in \mathcal{K}$ into $\mathcal{O}$ while fulfilling the capacity constraint $\sum_{s \in \mathcal{O}} g_a(\gamma_s)q_a(\mu_s)f_a(\omega_s) \leq \eta_{\text{o}}$ following the order.
5) Place $s \in \mathcal{J}$ into $\mathcal{O}$ while fulfilling the capacity constraint $\sum_{s \in \mathcal{O}} g_a(\gamma_s)q_a(\mu_s)f_a(\omega_s) \leq \eta_{\text{o}}$ following the order.
6) Place into $\mathcal{O}$ remaining $s \in \mathcal{K}$ and update penalties p_s.
7) $s \in \mathcal{O}$ will be notified as service requests accepted and $p_{\text{o}} = \sum_{s \in \mathcal{O}} p_s \left(g_a(\gamma_s)q_a(\mu_s)f_a(\omega_s)\right)$ will be notified to the 5G network management as $\lambda_{\text{o}} = \eta_{\text{o}} + p_{\text{o}}$.

It is clear that the priority threshold $\bar{\rho}$ is a key-parameter for driving the system towards optimal solutions. However, it can be a configurable parameter chosen by the network provider for guaranteeing different levels of QoS and, in other cases, identifying emergency situations. It can be also chosen based on different pricing models to foster external administrative domains to increase the pay-off for ensuring mission critical communications.

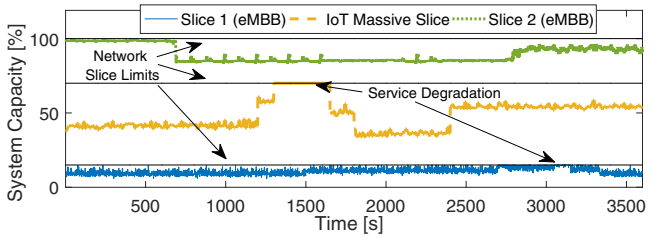
V. Performance evaluation

We carried out a preliminary simulation campaign to evaluate the compound effect of an IoT traffic-aware slice by means of an ad-hoc simulator, written in MATLAB®. In particular, we simulate a 5G network environment with 50 eNBs covering a 30 km^2 area. In this area, we deploy several IoT gateways collecting data from 50000 different sensors, including traffic sensors, air quality sensors, crowd control sensors and HD cameras, and transmitting data by means of 5G network facilities. The IoT deployment is structured onto $|\mathcal{S}| = 100$ not overlapping areas and, the service request per area might come to the IoT Broker at regular time interval equal to $n = 10$ min, with QoS configuration parameters randomly chosen. All simulation parameters are listed in Table II.

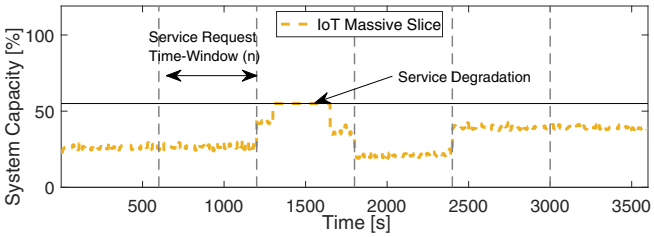
TABLE II: Simulation parameters

Area	30 km^2	**Capacity (UL)**	75 Mb/s
$\lvert\mathcal{S}\rvert$	100	**Capacity (DL)**	150 Mb/s
Sim. Duration	1 hour	**IoT service req. (n)**	10 min.
5G eNBs	50	**Quality levels** $\lvert\mathcal{Q}\rvert$	5
Data Frequencies $\lvert\mathcal{W}\rvert$	6	**Granularities** $\lvert\Gamma\rvert$	4
Service Priorities $\lvert\mathcal{P}\rvert$	5	**Priority Threshold** $\bar{\rho}$	3

We show the impact of implementing a joint IoT traffic brokering and 5G network slice traffic-aware mechanism compared with the legacy solution, where the 5G network management takes slicing decisions, independently. We assume that the 5G network management deals with three different slices, such as *Slice 1* characterized by enhanced Mobile BroadBand (eMBB) traffic, *Slice 2* and *IoT Massive Slice*,

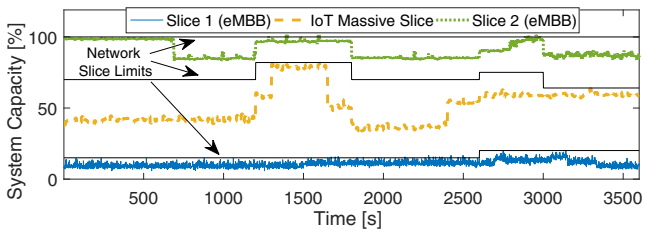


(a) Network Slices Traffic

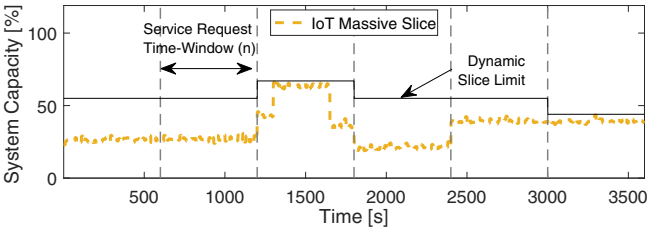


(b) IoT Massive Slice Traffic

Fig. 5: IoT Traffic adaptation without joint mechanisms



(a) Network Slices Traffic



(b) IoT Massive Slice Traffic

Fig. 6: IoT Network Slice Traffic-aware adaptation

asking for 15%, 30% and 55% of the overall system capacity, respectively. In Fig. 5, we show the system behavior when no joint mechanisms are devised. In Fig. 5(b), we focus on the IoT Massive Slice traffic by marking different time windows wherein IoT service requests vary. After the third service requests set, the IoT traffic dramatically increases and the IoT Massive Slice limit degrades the quality of service, as some service requests cannot be satisfied. Note that, also *Slice 1* experiences QoS degradation after 2800 seconds, as slice boundaries are fixed and not flexible.

In Fig. 6(a), we show the same system configuration after running our joint mechanism, where we implement the IoT Traffic-aware Slice Optimizer algorithm. In this example, the 5G network management can dynamically adjust the network slice limits based on the traffic dynamics. In particular, the IoT Massive Slice requires a slice resources upgrade at time 1200s because of some high-priority service requests. Since the other network slices are underutilized, the 5G network management can promptly accommodate this request, as shown in Fig. 6(b). When *Slice 1* requires more resources, the 5G network management triggers the IoT Broker thereby asking for a traffic reduction. This automatically leads the system to efficiently utilize available resources while avoiding service degradation.

VI. Conclusions

In this paper we proposed a novel system architecture in charge of efficiently creating and efficiently adjusting self-contained and isolated network slices in massive IoT scenarios building on IoT Brokers features. With our proposed solution, network operators and IoT tenants can interact to trade-off resources among slices to avoid service degradation. Our novelty relies on a new system architecture where 5G Network Slice Brokers and IoT Brokers are interconnected. Our results show that a joint IoT Broker/Network Slice Broker Orchestration might provide benefits to all parties. By means of simulations, we have shown that our *IoT Slice-aware Traffic Optimizer* algorithm can efficiently drive the system towards fully-utilization states while fulfilling the required SLAs.

References

[1] N. Alliance, "Description of the Network Slicing Concept," *NGMN 5G P1*, Jan. 2016.

[2] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From Network Sharing to Multi-tenancy: The 5G Network Slice Broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, July 2016.

[3] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile Traffic Forecasting for Maximizing 5G Network Slicing Resource Utilization," in *INFOCOM'17*, May 2017.

[4] C. Pereira, A. Pinto, D. Ferreira, and A. Aguiar, "Experimental Characterisation of Mobile IoT Application Latency," *IEEE Internet of Things Journal*, 2017.

[5] L. Toka et al., "A Resource-Aware and Time-Critical IoT Framework," in *INFOCOM'17*, May 2017.

[6] J. S. Leu et al., "Improving Heterogeneous SOA-Based IoT Message Stability by Shortest Processing Time Scheduling," *IEEE Transactions on Services Computing*, vol. 7, no. 4, pp. 575–585, Oct 2014.

[7] M. A. Nef et al., "Enabling QoS in the Internet of Things," in *CTRQ 2012: The Fifth International Conference on Communication Theory, Reliability, and Quality of Service*, 2012.

[8] S. Oh et al., "A Scheme to Smooth Aggregated Traffic from Sensors with Periodic Reports," *Sensors*, vol. 17, no. 3, 2017.

[9] N. C. Luong, P. Wang, D. Niyato, Y. Wen, and Z. Han, "Resource Management in Cloud Networking Using Economic Analysis and Pricing Models: A Survey," *IEEE Communications Surveys Tutorials*, 2017.

[10] C. A. Gizelis and D. D. Vergados, "A Survey of Pricing Schemes in Wireless Networks," *IEEE Communications Surveys Tutorials*, vol. 13, no. 1, pp. 126–145, 2011.

[11] L. Ramaswamy et al., "Towards a Quality-centric Big Data Architecture for Federated Sensor Services," in *2013 IEEE International Congress on Big Data*, June 2013, pp. 86–93.

[12] T. Jacobs et al., *D.14.2.2: FIWARE GE Open Specifications (IoT Chapter) - IoT Broker Release 5*. Future Internet Core, 2016.

[13] M. Richart, J. Baliosian, J. Serrat, and J. L. Gorricho, "Resource Slicing in Virtual Wireless Networks: A Survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, Sept 2016.

[14] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker, "Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks," *IEEE Communications Magazine*, May 2017.

[15] C. Papadimitriou et al., *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Inc., 1982.