Gazing into the crystal ball: when the Future Internet meets the Mobile Clouds

G. Piro, *Member, IEEE,* M. Amadeo, G. Boggia, *Senior Member, IEEE,* C. Campolo, *Member, IEEE,* L. A. Grieco, *Senior Member, IEEE,* A. Molinaro, *Member, IEEE,* G. Ruggeri, *Member, IEEE,*

***Abstract*—The latest advances in mobile devices and the widespread diffusion of networked objects are driving the evolution of traditional Mobile Cloud Computing (MCC) systems toward a new framework where storage, computing, sensing, and other device capabilities are offered *as a service* at the network edge. This visionary scenario, encompassing heterogeneous resources generated, shared, and consumed everywhere in the network, requires innovative architectural and protocol design. In this context, can the approaches recently formulated in the Future Internet research arena (e.g., middleware-based virtualization, Information Centric Networking, and Software-Defined Networking/Network Function Virtualization) support the evolution of mobile cloud systems? This paper provides an affirmative answer by proposing Future-MCC, a novel architecture that capitalizes on such promising approaches and re-thinks (when needed) their philosophy to better fit the evolution of MCC systems. The performance of Future-MCC has been investigated in a representative heterogeneous and dynamic Smart City scenario. Computer simulation results clearly demonstrate that the proposed solution ensures *(i)* a reduction of the bandwidth requirements spanning from 66% to 91% and *(ii)* an average energy saving equal to 99% with respect to a conventional cloud computing platform.**

***Index Terms*—Mobile Clouds, Future Internet, Information-Centric Networking, Middleware, Virtualization**

I. Introduction

For many years, *cloud computing* has gained popularity thanks to its native ability to offer an on-demand access to a shared pool of configurable computing resources (e.g., networks, servers, storage, software). Cloud resources can be flexibly provisioned and released with a minimal management effort, while reducing capital expenditures (CAPEX) and by decoupling services and underlying technologies [1]–[3]. With the widespread use of mobile devices (including smartphones, netbooks, and tablets) the traditional cloud computing has evolved toward the Mobile Cloud Computing (MCC) paradigm. MCC strengthens the capabilities of mobile devices with the support of remote data centers: complex services and applications can be executed without wasting local (typically constrained) resources (i.e., battery life, storage, bandwidth, CPU) [4], [5].

Today, the MCC landscape is experiencing a new change of perspective, driven by two key aspects:

- *The availability of cloud capabilities at the network edge*, that englobes personal devices able to share their resources with nearby nodes [6], [7].
- *The birth of the Cloud of Things concept*, according to which sensing and actuation features offered in the Internet of Things (IoT) domain can be abstracted, virtualized, and treated as cloud resources [8].

As a result, the evolution of MCC will deal with a quite complex ecosystem where billions of heterogeneous devices (e.g., mobile handheld devices, wireless sensors and actuators, cameras, connected cars, etc.) and data centers expose a large variety of local and remote *resources as services*. The term *resource* here takes a broad meaning: it may represent a content, a storage or a computational capability, the execution of a given action, a sensing operation, a networking capability, etc. Furthermore, the resources will be shared and consumed throughout the network, in a manner that is agnostic of their physical location, the configuration settings of the hosting devices and the communication technology.

Enabling this revolutionary scenario is inevitably a challenging task to accomplish. In fact, any practical implementation should take care of *(i)* the ability to discover, access, and manage heterogeneous and dynamically-available resources, also under intermittent and poor connectivity conditions, *(ii)* the optimization of network performance, and *(iii)* the provisioning of security services like user authorization and resource protection.

Many solutions in the Future Internet research arena are trying to provide (partially) answers to these issues. The most important approaches include:

- *middleware-based virtualization* [9], which targets the easy interoperability among heterogeneous low-layer technologies and applications;
- *Information Centric Networking (ICN)* [10], that aims to facilitate and secure data dissemination;
- *Software-Defined Networking (SDN)* [11] and *Network Function Virtualization (NFV)* [12], that provide a more flexible and programmable utilization and management of network resources and functions.

It is surprising that, at the time of writing and to the best of our knowledge, all of these promising solutions have not been harmonized yet in a single framework that could fully satisfy the requirements of upcoming MCC scenarios.

To bridge this gap, herein we present a novel architecture, namely *Future-MCC*, which integrates, orchestrates, and further enhances the aforementioned Future Internet solutions.

G. Piro, G. Boggia, and L. A. Grieco are with the Dep. of Electrical and Information Engineering (DEI), Politecnico di Bari, v. Orabona 4, 70125, Bari, Italy; e-mail: {name.surname}@poliba.it.

M. Amadeo, C. Campolo, A. Molinaro, and G. Ruggeri are with Universita' Mediterranea di Reggio Calabria, Reggio Calabria, Italy. e-mail: {name.surname}@unirc.it

In order to assess the effectiveness and the efficiency of the proposed architecture, a Smart City scenario has been modeled and evaluated through computer simulations. Obtained results clearly demonstrate that Future-MCC significantly outperforms a conventional cloud-computing platform, both in terms of bandwidth requirements and energy consumption. Accordingly, Future-MCC positions itself as a very promising deployment solution for upcoming MCC architectures.

The remainder of the paper is organized as follows. Emerging trends of future MCC are summarized in Section II. The reference scenario considered in this paper and related open issues are discussed in Section III. Section IV provides background materials of approaches proposed in the Future Internet research arena, that can be adopted to enable future MCC systems. Section V presents a detailed description and the main features of the proposed architecture. Performance evaluation is reported in Section VI. Finally, Section VII draws conclusions and hints for future research.

II. Mobile Cloud Computing: upcoming trends

Recently, computing and networking technologies have evolved at an incredible pace, on the one hand, to match the growing user demands and, on the other hand, to fully exploit the increasing capabilities of personal devices and resource-rich network nodes. They are undergoing groundbreaking paradigm shifts, having in *data-driven* approaches and *virtualization* the main drivers.

Cloud computing provides computing and storage services, and different kinds of applications over the Internet. Conventional cloud platforms are deployed on complex and distributed data centers and hide the aforementioned capabilities behind a remote entity, whose physical location and configuration are typically unknown to end users [2].

Research rapidly evolved towards extending cloud computing capabilities offered by remote data centers to *mobile devices*. This is the target of the MCC paradigm [13], which intends to empower mobile devices to run a wide range of applications with increased complexity, such as games, image/video processing, e-commerce, and online social networks, without exhausting their limited resources [4], [5]. Remote cloud capabilities can be also exploited to process/store the massive amount of real-world data coming from the multitude of sensors embedded in today's mobile devices to support, for instance, health/fitness and environment monitoring applications [14].

In addition to such a perspective, typically referred to as *infrastructure-based mobile cloud* [5], the concept of *local mobile cloud* has been developed. It refers to a group of mobile devices that acts as a cloud and provides access to local services to other mobile devices. This is the case of vehicles leveraging their underutilized resources and playing the role of a cloud within which services are produced and consumed [15].

Better performance are expected by combining local and remote resources provided, respectively, by mobile devices and remote data centers, as envisioned in the *Cloud 2.0* proposal by AT&T [7]. This approach could overcome the limitations of an exclusively cloud-centric model, by providing the following advantages: *(i)* reducing the traffic in the backbone network through a wise local versus remote load balancing, *(ii)* improving the satisfaction of end users, who can benefit from a reduced monetary cost for accessing remote cloud facilities, and from faster and more flexible service access, and *(iii)* cost savings for producers, who can reduce CAPEX.

In parallel with the drift of the cloud computing paradigm from remote data centers to mobile devices, the IoT has represented an important shift in the IT market. It is a key enabling paradigm for billions of networked objects offering services (mainly sensing, monitoring and actuation) in application domains that range from smart home, to smart transport, smart city, smart grid, etc. [16].

The simple integration between cloud computing and IoT allows sensors to provide their sensed data to a storage cloud service, which then undergoes data analytics tools for knowledge discovery [17]. However, complementing IoT with data storage and processing capabilities should not be the only role of cloud in this context; cloud concepts should rather inspire the manner in which IoT resources are provided. Such an integration, commonly referred to as *Cloud of Things*, is fostering the use of virtualization techniques in the IoT domain. IoT capabilities and data are exposed in the form of services, in the same way as shared cloud resources are provided to computers and other devices *as an utility*. Virtual resources are easier to manage because they expose a uniform interface through standard abstractions. They can be easily shared if too big, and composed if too small. Virtualization enables multi-tenancy, by allowing re-usability of sensor information for a variety of applications and exposing a common interface to application developers.

Such appealing trends result in a novel landscape for MCC, whose bounds are unknown, and that raises a lot of unprecedented issues as discussed in the next Section.

III. Reference scenario and related requirements

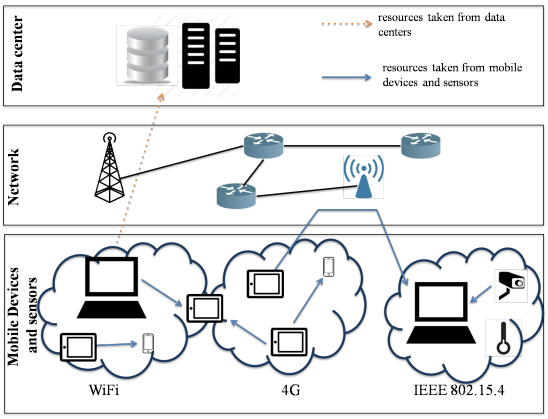


Fig. 1. Reference scenario: heterogeneous devices consume and share their local resources and access resources remotely provided.

The novel contribution of this paper with respect to the aforementioned approaches mainly stands in extending cloud computing capabilities to the end devices in the access segment

of the more complex scenario illustrated in Figure 1, where capabilities of mobile devices and IoT functionalities are offered and consumed locally. The resulting environment is expected to be extremely *heterogeneous* and *dynamic*.

The heterogeneity is a multi-dimensional feature and refers to:

1) *resource types*: the reference scenario embraces different kinds of resource categories, including data, storage, computing, sensing/actuation functionalities, and so on. Resources belonging to the same category may offer different capabilities; think, for example, of the storage space or the processor speed;
2) *applications*: a large number of software instances leverages available resources for providing end users with advanced services ranging from entertainment and gaming to environmental monitoring;
3) *devices*: the network holds a myriad of devices that differ in terms of hardware and software [18]. They include resource-constrained nodes (with limited energy, storage, and processing capabilities), multi-faceted mobile equipments (such as mobile phones and connected cars), and remote data centers (offering much larger hardware and software capabilities). Devices may offer raw data (like those measured by sensing units) or content formatted according to a specific standard, they can expose their local memory and/or processing resources.
4) *communication technologies*: devices can be attached to the access network and reciprocally interact over heterogeneous links. Communication technologies encompass broadband (e.g., 4G, 5G, Wi-Fi, etc.) and short-range low-power (e.g., Bluetooth, ZigBee, etc.) wireless networks, and also wired network segments [4].

The dynamicity, instead, covers both *resource status* and *network conditions*. Indeed, the availability of resources could be tremendously variable and subjected to the status of devices (i.e., mobility, battery, load, duty-cycle, etc.). Network conditions may also vary in time and space, e.g., according to the traffic load experienced over the access and core segments, due to congestion and radio propagation effects [4].

Starting from these premises, the provisioning of advanced (and distributed) services brings to a number of challenging requirements as discussed below.

Interoperability. A full decoupling between high-level applications and physical resources is needed to manage the inherent heterogeneity and complexity. Resources should be accessed by hindering all the header details. To this aim, a valid methodology is needed to describe the main characteristics of available resources by means of standardized interfaces.

Scalability. The envisioned scenario will be plagued by the explosion of data/signaling traffic generated by billions of devices (up to 25 billions in 2020 according to Cisco forecasts [19]) massively requesting access to a wide range of resources. Scalability should be wisely addressed. Multicasting could be an option when a given resource is requested by multiple nodes. Local interactions among devices at the network edge could be enforced and facilitated, regardless of the underlying communication technology, in order to offload the core network. In addition, name-based techniques could be strategic to ease the classification and the identification of resources in a way that is agnostic of the location of resources themselves. The location of resources, in fact, could be unknown a priory by the end-points, or irrelevant to them.

Robustness. Due to the dynamicity of the environment, users may experience service interruption, e.g., due to mobility, sleep operations or unreliable links. Thus, the information and communication technology services must be resilient to system failures, e.g., asynchronous communications should be supported.

Adaptability. For an efficient utilization of distributed resources, the entire network should be dynamically configured. Unexpected dynamics should be considered that characterize both physical and virtual topologies (e.g., congestion over crowded links and losses over the wireless segments), services requirements, and the Quality of Service (QoS) experienced by end users. In addition, to encourage innovation, broader flexibility is required to accommodate the natural evolution of services, technologies, requirements, and resources.

Security. The heterogeneity and the wide scale of the envisioned ecosystem magnify security threats that affect both (mobile) cloud computing [4], [20] and IoT [21], [22]. In general, the access to resources must be protected against un-legitimate users. Nevertheless, connection-oriented security schemes, widely used in the current Internet, could not be the best fit for the needs of the considered scenario [22].

IV. Enabling technologies

Research in the Future Internet context [23] has recently formulated some valid concepts that exhibit promising baseline capabilities for the envisaged scenario. They include middleware-based virtualization, ICN, and SDN/NFV. While an immediate overview is provided in Table I, their core functionalities exploited and extended in our proposal will be discussed below.

A. Middleware-based virtualization

At the time of this writing, *middleware*-based virtualization is considered as the most suitable solution to ease the usage of heterogeneous resources in distributed environments and to bridge the gap between the high-level requirements of the applications and the low-level hardware complexity [9]. Available approaches (e.g., Hydra, ASPIRE, UBIWARE, SOCRADES, and SIRENA [9]), focus on different aspects, such as device management, interoperability, platform portability, context-awareness, security and privacy, and many others. More recently, instead, the European Telecommunications Standards Institute (ETSI) has released a set of specifications defining a RESTful architecture to standardize the way heterogeneous devices can offer services and access seamlessly [24], [25], [26], thus easing Machine-to-Machine (M2M) communications. According to the ETSI M2M proposal, resources are uniquely addressable and identifiable via a Uniform Resource Identifier (URI).

Definitely, middleware-based virtualization allows to share, describe, and retrieve resources in a unified, effective, standardized, and deployment-independent manner. Thus, it could

TABLE I
MAIN FEATURES OFFERED BY EMERGING FUTURE INTERNET TECHNOLOGIES

Enabling technologies	Feature	Description
Middleware-based virtualization	Standardized resources description	Resources are exposed through standardized interfaces, bridging the gap between the high level requirements of the applications and the low level hardware complexity.
Information-centric networking	Content-centric communications	The core of the communication is the content (i.e., the resource) that can be shared, discovered, and delivered within the network.
	Naming	Resources are identified by unique names, allowing applications to retrieve them without any awareness about the physical location of servers (e.g., IP address). Names may range from flat to hierarchical and may (or may not) be human-readable.
	Publish/subscribe mechanism	Producers publish resources to which interested consumers may subscribe. This mechanism allows publication and subscription operations to be decoupled in both time and space domains (e.g., supporting mobility).
	Native multicast support	ICN nodes can identify requests for the same named information, avoiding the need to forward them differently on the same path.
	Routing-by-name	Requests are forwarded toward the closest destination(s), based on the names of the desired content, instead of relying on the IP address of the host. Corresponding replies are sent back to the sender through the reverse path.
	In-network caching	Each network entity traversed by a content packet destined to a consumer may decide to cache it, according to the adopted caching technique, by reducing server workload and saving bandwidth resources.
	Name-based security	It is possible to encrypt and authenticate directly names and contents, without relying on sophisticated schemes conceived instead for protecting the communication channel among nodes.
SDN/NFV	Separation of control and data planes	Network intelligence taken out of network nodes and placed in logically centralized controllers to allow flexible management, configuration, programmability of the network.
	Decoupling of physical network equipment from their functions	Virtual network functions implemented in software allow their instantiation at different network locations without necessarily requiring the purchase/installation of new hardware.

cover a key role in meeting the requirements related to resource and application heterogeneity and coexistence of different communication technologies.

B. Information Centric Networking

ICN is emerging as a promising networking paradigm for the Future Internet [10], which aims at evolving the classic host-centric Internet design to better support nowadays applications [27]. It has been or it is currently investigated and developed in several projects, such as Publish Subscribe Internet Routing Protocol (PURSUIT), Name Data Networking (NDN), and MobilityFirst, just to name a few [28] [29]. Despite some distinctive differences (e.g., content naming schema, security-related aspects, routing strategies, and cache management), they all share a receiver-driven communication model, based on content names and in-network caching [30].

With ICN, resources can be addressed through names that do not contain any reference to their publisher's location: this is crucial to enable advanced discovery mechanisms and data-centric resource sharing (especially among local devices). ICN also offers sophisticated mechanisms to trust exchanged contents without requiring to initialize a secured communication link with the publisher. Moreover, through built-in in-network caching and publish/subscribe interactions it can manage dynamically available resources and intermittent connectivity as well as provide a native support of multicasting.

The adoption of ICN in IoT scenarios is discussed in [31] and [32]. In these contributions, naming schemes are extended to support not only data retrieval but also IoT sensing and actuation services, and in-network caching is rethought to better match the characteristics of IoT contents. In fact, unlike Internet contents, IoT sensing data are usually *transient*, i.e., they frequently change in time. They would be cached only for a limited period and updated with fresher values. For actuation operations, instead, the output would be cached if it is of potential interest for several nodes and only provided to the requesting consumer otherwise.

Without loss of generality, our proposal builds upon NDN [33], one of the most promising and popular ICN architectures, characterized by a highly flexible and robust communication model, fitting both fixed and dynamic environments. By relying on *hierarchical* names, NDN represents a good candidate to facilitate interactions with middleware facilities naming resources with URIs. NDN communication is based on the exchange of two kinds of packets: the *Interest*, used to request a resource, and the *Data*, used to provide a corresponding answer. The Data also embeds security information (e.g., the signature of the producer), thus integrity and trust travel with the content itself [34]. Each NDN node maintains three data structures: *(i)* the Content Store (CS) that temporary caches incoming Data, which can be used to satisfy future requests; *(ii)* the Pending Interest Table (PIT) that keeps track of the forwarded Interests and the interface they arrived from, thus Data can flow back to the requester(s); and *(iii)* the Forwarding Information Base (FIB), used as a routing table to select the outgoing interface(s) for incoming Interests.

In Section V, we will show how such NDN features can be merged in our proposal to support an efficient discovery of heterogeneous resources.

C. Software-defined networking/Network function virtualization

SDN and NFV are widely deemed two critical and complimentary pillars of the future Internet. SDN provides network programmability, by decoupling the control and data planes [11]. It allows for a control intelligence (control plane) that is logically centralized and decoupled from network devices (data plane). It can rely on a global network view, including information about the network topology, the traffic statistics and network usages. Fed by this closed loop control, adaptive packet forwarding rules can be defined to augment the overall system performance, i.e., offloading of the network, improvement of the QoS experienced by end users, etc.

NFV is the concept of transferring network functions from dedicated hardware appliances to software-based applications, for example to allow them to be hosted on virtual resources, e.g., a Virtual Machine (VM), of server platforms in cloud data centers [12]. The extreme dynamism and complexity of the envisioned scenario make the management of the whole system highly complicated and require to borrow from SDN and NFV to ensure flexible network resource usage.

SDN could facilitate data plane redirection mechanisms to allow designated traffic to reach the intended destination (e.g., the proper remote data center facilities, the nearest producer in the local cloud, the freshest sensed data, the closest content). Routing paths may be selected and, even, pre-configured by the SDN controller, according to the requirements of delivered data (e.g., bandwidth reservation guarantees, delay sensitiveness). The functionalities of the SDN controller (e.g., traffic monitoring, load balancing) can be implemented as Virtual Network Functions (VNFs) [12], to realize better service agility.

V. The *Future-MCC* Architecture

The proposed *Future-MCC* is a high-level and general-purpose architecture, supporting a wide set of advanced and heterogeneous services in future MCC systems. It intends to ease and optimize the global sharing of resources within the network, while meeting interoperability, scalability, robustness, adaptability and security requirements pinpointed in Section III. To this purpose, it leverages the functionalities and features of the future Internet paradigms discussed in the previous Section.

As depicted in Figure 2, *Future-MCC* consists of five conceptual planes:

- **The *Cyber Physical* plane** integrates heterogeneous resources which are used according to the publish/subscribe mechanism and for which a *virtual* representation is generated by the middleware. The latter one also offers standardized interfaces, e.g., software Application Programming Interfaces (APIs), through which exposing a resource to the rest of the architecture (publish) or to communicate the interest to access to a given resource (subscription), as well as to control the access to resources and to provide their protection against unauthorized users.
- **The *Service* plane** hosts both *resource producer* and *resource consumer* applications. The resource producer initiates the sharing of resources available on the device and monitors their usage during the time. From another side, the resource consumer is the entity of the proposed architecture that would use a set of resources. Note that all of these functionalities are executed by leveraging APIs made available by the middleware layer. That is, the resource producer uses the middleware for exposing resources in a standardized way; the resource consumer uses the middleware for handling the (controlled) access to remote resources.
- **The *Network* plane** acts as the *communication bus* that interconnects all the virtual resources available in the Cyber Physical plane and makes them easily accessible from different nodes. Network nodes are part of an ICN overlay infrastructure that hides the heterogeneity of underlying radio and wired access technologies (ZigBee, WiFi, WiMAX, cellular, Ethernet, etc.). Such an overlay brings to the definition of logical links through which heterogeneous nodes interact by using ICN primitives.
- **The *Control* plane** hosts the Traffic Engine entity that optimizes and dynamically configures the network plane.
- **The *Management* plane** embraces two entities, i.e., the Service Engine and the Authorization Server. The former one provides resource mapping and service composition operations. The latter one, instead, implements security functionalities.

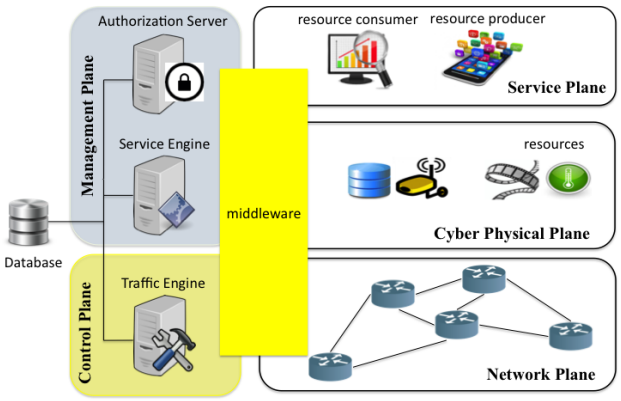


Fig. 2. Big picture of the conceived architecture.

A. The middleware

What immediately emerges from the big picture of *Future-MCC* shown in Figure 2 is that the middleware plays a crucial role in the entire system. All the functionalities it offers, in fact, are extremely useful to satisfy the requirements of upcoming MCC architectures. In few words, the middleware provides a *unified communication language* between different applications that aims at exposing heterogeneous resources (or accessing to them) by means of a wide range of underlying communication technologies. To reach this goal, any entity of the conceived architecture (not only resource consumers and resource producers, but also logical entities of the *Management*

plane and nodes belonging to the *Network* plane) runs an instance of the middleware layer that implements specific functionalities and interacts with the rest of the architecture through standardized interfaces. Therefore, *Future-MCC* leverages a distributed middleware.

Many devices in *Future-MCC* (e.g., IoT nodes) may exhibit resource constraints. By inheriting the design criteria of IoT systems already presented in ETSI-M2M specifications [24] and in the context of the Internet Research Task Force (IRTF) Information-Centric Networking Research Group (ICNRG) [35], *Future-MCC* assumes that such devices are hidden behind a single gateway device. The gateway implements a middleware instance and executes registration operations on their behalf and performs local discovery tasks. Note that the middleware instance running on the gateway device may also offer *Context Aware Processing* (CAP) capabilities: data generated by resource producers (think for instance to a temperature value measured by a sensor node) can be processed and stored at the gateway side. In this way, the middleware instance in the gateway is able to directly satisfy multiple requests for the same resource without needing to contact the resource producer many times. Therefore, if properly configured, this mechanism can significantly reduce the amount of energy consumed by the resource producer.

In summary, the list of APIs made available by the middleware are:

- **Register**, used by a producer to expose its own resources in the architecture;
- **Request**, used by a consumer to retrieve a set of resources of interest;
- **Subscribe**, generally adopted to make a subscription to a virtual resource (including information of the *Network* plane to be processed by the Traffic Engine entity);
- **Configure**, exploited by the Traffic Engine of the *Control* plane to optimize and dynamically configure the ICN overlay;
- **Traffic Update**, used by nodes of the *Network* plane to provide updates on the traffic load to the Traffic Engine of the *Control* plane;
- **Service Update**, used by the resource consumer to communicate its experienced QoS to the Service Engine of the *Management* plane.

Their practical usage is, instead, described in the following subsections.

B. Registration of virtual resources

The registration of a virtual resource is performed in three consecutive steps. First, a producer application that wants to expose a resource available on the device issues a *Register* request to the middleware instance. Then, the middleware generates a standardized representation of such a resource. Finally, the availability of the resource, along with its standardized representation, is communicated to the *Management* plane.

Note that the standardized representation of a resource contains three main fields: *meta-data*, *name*, and *locator*. The *meta-data* field stores all the details associated to a given resource, including type, amount, and availability information. Such details are then summarized within the *name*, which identifies the resoure in a hierarchical name-tree. The same *name* can be assigned to multiple resources that, even being exposed by different producers, have the same set of high-level properties. The resource *locator*, instead, provides the exact position of the resource producer, in terms of network address. *Meta-data* are used to support resource mapping and service composition operations, as described in Section V-C. *Name* and *locator*, instead, are used during distributed and centralized discovery operations, respectively, as discussed in Section V-D.

Just to provide an example, Figure 3 shows a possible standardized description of a video camera.

```
<description>
  <meta-data>
      <type> camera </type>
      <resolution> 1028p </resolution>
      <codec> x264 </codec>
      <location> Country/City/Street </location>
  </meta-data>

  <locator> URI </locator>

  <name>
  /camera/Country/City/Street/camera_1028p/codec_x264/
  </name>

</description>
```

Fig. 3. Example of a standardized description related to a camera resource.

C. Resource mapping and service composition

In general, the resource consumer does not know, a priori, the exact set of resources needed to execute a given service. Furthermore, multiple resources may exist in the envisaged MCC scenario, which can be leveraged to accomplish a given service.

Hence, the resource consumer issues a high-level request by using the *Request* API of the middleware. For instance, it could request the provisioning of a high-quality video surveillance service in a given road area. The request is sent to the *Management* plane, which is responsible for the identification of virtual resources (i.e., a camera in the surveillance example) that can be discovered (first) and retrieved (then) for completing the execution of a given service.

The logical node of the *Management* plane involved in this task is the Service Engine. It mainly executes two parallel tasks, that are *resource mapping* and *service composition*. These operations are jointly executed to identify the kind of resources requested by the service, as well as to combine multiple basic resources for supporting a composite high-level service. Note that the service composition task is extremely important especially in the IoT context, where the limited capacities of constrained devices can be properly exploited in more complex, aggregated, and coordinated functionalities.

Then, the Service Engine will generate a response message containing *meta-data* and *names* related to virtual resources to discover. This message will be received by the middleware instance running on the device of the resource consumer and processed at the application layer. Thanks to the information

stored in the *meta-data* field, the resource consumer learns all the details required to effectively use the resources.

D. Resource discovery

As soon the resource consumer receives the answer to its high-level request, it can subscribe to these resources through the *Subscribe* API offered by the middleware.

The resource consumer knows the details of these resources and, for this reason, it already knows the type of data that can be fetched from them. However, it has no idea about the position/identity of resource producers. The discovery process is, in fact, handled by the middleware in a way that is transparent to the upper layer application. This is done through a hybrid mechanism, which jointly integrates a *distributed* and a *centralized* approach. Since the resource availability and location is not known in advance, the middleware instance running at the consumer side enforces the distributed discovery first, and in case of a failure (i.e., the requested resource cannot be found locally), it triggers the centralized discovery after a timeout expiration[1].

Distributed resource discovery. Such an approach has the advantage to quickly discover a resource, when it is locally available, without overloading the network and the remote entities. In our architecture, it builds upon ICN, by exploiting NDN primitives. The middleware issues an *Interest* packet carrying the name that identifies a virtual resource exposed by the *Cyber Physical* plane. The Interest will be routed within the ICN overlay. Intermediate nodes forward the Interest packet by looking up the name in their FIB, which are populated by the Management plane. Once a node able to satisfy the request (i.e., a device storing the desired content or able to execute the requested actuation/sensing task, etc.) is reached, it is triggered to reply with a corresponding *Data* packet. The latter one may carry either a content payload (if a content resource has been requested) or the outcome of the performed computation/action (otherwise). In both cases, it can also integrate authentication information. Data packets are returned based on the state information set up by the Interests at each traversed node. They can be cached by traversed nodes, e.g., if they either carry a content or a processed output.

Centralized resource discovery. The centralized approach is the most robust solution to discover resources that can be only remotely provided, either because the consumer has no short-range connectivity to producers in the local cloud or because huge resources are needed for the provisioning of the desired service.

In such a case, the middleware instance sends its request to the Service Engine belonging to the *Management* plane, that will provide the *locator* of the most suitable resources (as identified by the resource mapping and service composition tasks) to which releasing the subscriptions. Once the set of required resources has been retrieved, consumer and producer applications will interact by using specific network protocols (which may differ from the aforementioned ICN communication paradigm).

[1]The timeout setting is the result of the tradeoff between the network load and resource retrieval delay.

E. Network optimization

The Traffic Engine of the *Control* plane is in charge of enforcing sophisticated strategies aimed at dynamically optimizing network operations, by leveraging the (global) knowledge about the network and available resources. To this end, the resource consumer and network nodes uses *Service Update* and *Traffic Update* APIs of the middleware to provide feedbacks related to the QoS level experienced by the end user and the traffic load registered at the *Network* plane, respectively. Starting from these details, the Traffic Engine uses the *Configure* API of the middleware to set the properties of the *Network* plane, i.e., by dynamically defining the topology, routing algorithms, and parameters for both the physical communication infrastructure and the overlay ICN network.

Just to provide an example, the *Control* plane can successfully build the FIB tables of ICN nodes starting from the knowledge of the resources locations, either by updating the related FIB entries after having detected anomalous conditions or congestion events, or by configuring the path rules to meet the application priority (e.g., multiple paths toward a given resource for reliability purposes; or a single path toward the closest/less loaded resource for a time-bounded emergency application).

Hence, the network optimization consists in the definition of optimal (or alternative) paths toward remote resources, as well as in the reduction of the overhead generated by common routing algorithms. From the implementation point of view, such capabilities are addressed by integrating high programmability and flexibility features belonging to the SDN philosophy [36]. Moreover, in order to counteract the *single point-of-failure* issue and run complex tasks (e.g., multi-criteria optimization routing algorithms on a large scale), the *Control* plane leverages robustness and redundancy capabilities offered by data centers. Of course, centralized routing schemes may lead to long convergence delay due to the fact that a sudden failure must be first reported to the *Control* plane, which recompiles the routes and then disseminates updates. In this regard, adaptive forwarding schemes can be integrated in the *Network* plane as an additional feature aimed at reducing the convergence delay [37]. Moreover, they provide a valid alternative when the connectivity with remote *Control* plane entities becomes unavailable (e.g., under high mobility conditions).

F. Security support

Security is another big concern for MCC architectures [20]. Unfortunately, connection-oriented security services, widely used in the current Internet, may poorly fit the requirements of upcoming MCC architectures [22]. Due to the distributed access to virtual resources and the constrained nature of the majority of resource producers, in fact, conventional secured schemes bring to serious scalability issues. To solve this problem, *Future-MCC* integrates and extends some valid solutions recently developed in the literature, like [22] and [38], thus offering the protection of resources against unauthorized accesses. The main idea is that the *Management* plane hosts a

trusted Authorization Server that orchestrates security services by introducing novel features to both resource publishing and retrieving functionalities. The resource protection is reached by limiting the access to only a set of authorized users. To this end, the publication and the retrieving of virtual resources is executed as in the following (see Figure 4):

- the resource producer triggers the registration of its resources by issuing *Register* messages to the middleware;
- the middleware generates a standardized representation of such resources, establishes a secure connection with the Service Engine of the *Management* plane, and sends to it the descriptions of these resources;
- the Service Engine contacts the Authorization Server for setting up security services;
- in line with [22], the Authorization Server assigns to each resource a *secret*, S, which is feedback to the middleware entity of the resource producer;
- when a resource consumer issues a high-level request, its middleware instance establishes a new secured connection with the Service Engine;
- before providing the result of resource mapping and service composition processes, the Authorization Server is contacted by the Service Engine for verifying that the resource consumer may access to a given set of resources. This task can be managed through the authorization mechanisms already proposed in [38];
- if the authorization process ends with success, the Service Engine generates an *access token* for each of the selected resources. The token, A_i, is a message encrypted with the secret associated to that resource and containing the identifier of the resource consumer: $A_i = E_{S_i}[t, u]$, where $E()$, S_i, t, and u represent the encryption operation, the secret assigned to the i-th resource, the timestamp and the user identifier, respectively;
- during the resource discovery, the middleware appends in each request the *access token* associated to the resource it wants to retrieve;
- once the request is received by the middleware instance of the resource producer, the validity of the *access token* is firstly verified. The corresponding data is generated only if the previous control does not generate any error.

Note that all the aforelisted security operations are performed by the middleware layer. In the case a constrained device exposing a given resource is connected to the *Future-MCC* platform through a gateway (as already explained in Section V-A), the protection of resources against unauthorized accesses is directly handled by the gateway itself. Therefore, no (heavy) operations are assigned to constrained devices. In addition, our proposal assumes that the communications in the IoT domain are secured. Even if the way this goal is reached is out-of-scope of the proposed work, some standardized techniques or solutions proposed in the literature (see for instance [39] and [40]) could be used in this context.

VI. Performance Evaluation

The effectiveness of *Future-MCC*, as well as the performance gain offered with respect to a conventional cloud computing platform, have been evaluated through computer simulations. In particular, the retrieval of resources is investigated at the network-level perspective, thus studying both bandwidth requirements and energy consumption.

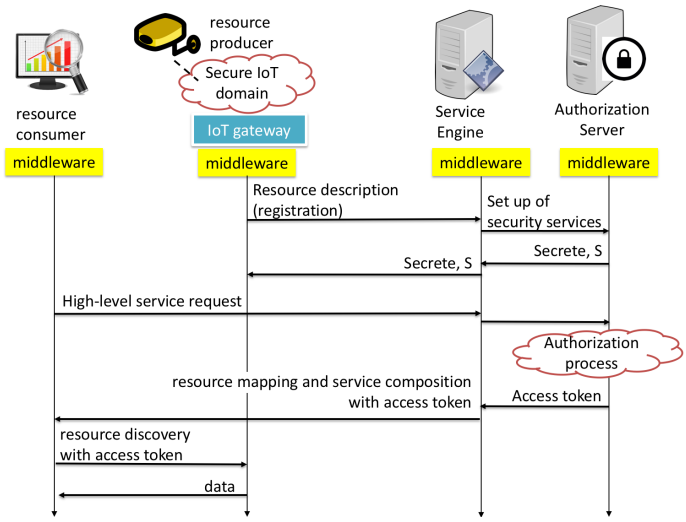


Fig. 4. Sketch of the security model designed for *Future-MCC*.

A. Reference scenario, performance metrics, and benchmarking solutions

A Smart City is a representative scenario encompassing the provisioning of advanced and composite services (including smart transportation, environment monitoring, surveillance). It integrates devices that are highly heterogeneous in terms of hardware/software capabilities, communication interfaces, mobility patterns, and offered functionalities.

The service considered in our work is the retrieval of sensing data from monitoring and surveillance applications. Specifically, resource consumers (i.e., public authorities, citizens, fire brigades) are interested to get an updated and detailed picture of what is going on in a given portion of the city. Note that this service is quite common in a smart city, e.g., to build the scene of a given road segment after an accident, to prevent a terrorist attack, or to assess the damages and to coordinate the aids after a natural disaster.

Resource providers, instead, include:

- *Devices with sensing capabilities*: they form a very large group of nodes embracing webcams and sensors. Such purpose-built devices expose sensing functionalities to monitor the environment (e.g., air pollution, noise), or to provide traffic efficiency/surveillance applications (e.g., measuring congestion, monitoring road conditions). Depending on the deployment strategy, devices with sensing capabilities could be managed by gateways able to expose their resources in a standardized manner and to offer *context-aware* processing capabilities.
- *User devices* (e.g., smartphones): they typically provide user-generated contents, like data, pics, and videos, that can be shared and used by other users/entities during the execution of advanced services and serve crowdsensing applications.

Two main performance indexes have been evaluated: the bandwidth requirements (i.e., the amount of data exchanged among resource producers, remote cloud platform, and resource consumers in both access and core networks) and the energy consumption at the producer side (i.e., the amount of energy spent for handling the aforementioned data exchange). In this preliminary work, the bandwidth requirements are calculated by only considering the amount of data generated at the application layer. The overhead introduced by the protocol stack and control messages have been neglected. For what concerns the energy consumption, results will be reported in terms of *unit of energy*, defined as the amount of energy needed to transmit a single byte of information.

A conventional cloud computing platform is considered as a baseline approach. In this case, all the resource producers push their data to the remote data center, where they will be stored, processed, and made available for consumer applications. While this task is always done, even if there are not consumers interested in such resources, the remote cloud platform acts as the bottleneck of the entire system. As a result, the consequent high communication and computational load will make cloud-centric solutions not scalable with the number of producers and consumers of resources. *Future-MCC*, instead, overcomes such issues. In fact, contents are sent to resource consumers only when they are requested, and all the services are natively offered in a distributed way. First, the direct retrieval of distributed resources would immediately gain benefits in terms of bandwidth requirements. Moreover, the adoption of *Context Aware Processing* functionalities may also ensure a significant reduction of energy consumption at producer devices that are typically resource-constrained.

B. Simulation models and parameters

The performance of Future-MCC has been evaluated through a customized simulation framework that implements abstraction models for the core network, the resource consumers and producers, the relevant request/generation patterns, and the remote cloud platform. Assumptions and methodologies are presented in the following. For the sake of clarity, a summary of all the adopted symbols is reported in Table II.

TABLE II
LIST OF MAIN NOTATIONS.

Parameter	Description
N_p	Total number of composite services [#]
N_c	Total number of resource consumers [#]
V_n	Number of cameras producing video contents for a given composite service [#]
V_s	Average size of video contents [bits]
V_r	Rate of video contents requests [req/s]
S_n	Number of sensing devices producing data for a given composite service [#]
S_s	Size of sensing data [bits]
S_r	Rate of sensing data requests [req/s]
U_n	Number of mobile devices providing user-generated contents for a given composite service [#]
U_s	Average size of user-generated contents [bits]
U_r	Rate of user-generated contents requests [req/s]

The number of resource consumers is set to N_c. Each consumer asks for a composite service, which requires the access to three different kinds of resources: video recordings, sensor data, and user-generated contents. N_p represents the total number of composite services. Specifically, a composite service is made up of V_n cameras producing video contents, S_n sensing devices, and U_n mobile devices providing user-generated contents. Moreover, V_s, S_s, and U_s are the average size of video, sensing, and user-generated contents, respectively. For simplicity, these variables take also care of the overhead due to the protocol stack. V_r, S_r, and U_r variables, instead, are used to indicate the frequency of resource requests from the consumers.

In our tests, we set $V_n = 2$, $S_n = 100$, and $U_n = 2$. The S_n setting well reflects the need for a big amount of sensing data for accuracy purposes (many parameters of the same type can be collected) and for a better scene representation (several types of sensed data may be required, e.g., air pollution, noise). A video content is modeled as a source that generates frames at a variable bit rate (the average encoding rate is set to 128 kbps). As a consequence, $V_s = 5120$ bits and $V_r = 25$ req/s. Sensor data and user-generated contents, instead, are requested every second (i.e., $S_r = 1$ req/s and $U_r = 1$ req/s). Sensing devices generate data with a constant size $S_s = 100$ Bytes. The size of user-generated contents is modeled through a geometric distribution with average size $U_s = 100$ Kbits.

Similarly to web-based contents, we assume that the popularity of a composite service follows the Zipf distribution [41]:

$$P(i) = \frac{i^{-\alpha}}{\sum_{j=1}^{N_p} j^{-\alpha}}, 1 \leq i \leq N_p, \quad (1)$$

where $P(i)$ and α are the popularity of the $i-th$ composite service and the Zipf coefficient characterizing the popularity profiles of all the available services, respectively. N_p, instead, is the number of composite services. In every run, consumers ask for a given composite service, based on the probability distribution defined before. To properly model their preferences in the case of user-generated contents, we set $\alpha = 0.9$.

The simulated network reproduces the GEANT topology, which is composed of 22 nodes at the core network, organized as depicted in Figure 5 [42]. They are the attachment points for resource producers and consumers. Moreover, to evaluate the impact that the traffic load has on both bandwidth requirements and energy consumption, we stress the network by setting $N_p \in [200 - 1000]$ and $N_c \in [0.1N_p - 0.5N_p]$. Without loss of generality, it is supposed that all the resources belonging to a single composite service are physically attached to a gateway device, connected to a specific *network attachment point* (i.e., the access point of a WiFi network, the base station of a cellular system, etc.)[2]. In each run, the network attachment points of both resource consumers and resource producers are randomly chosen among those belonging to the simulated GEANT network. For the baseline approach, the cloud platform is abstracted as a repository where resource producers push their data. To ease the interpretation of results,

[2]Given the considered use case, this assumption is not far from the reality: resources that are of interest for a user may be geographically located in the same limited area, where the Internet connectivity is offered by a specific network access node.

we assume that resources offered by the cloud can be reached through a single network attachment point (i.e., a router of the core network) over the GEANT topology (see Figure 5).

In the developed simulation framework, routers of the core network, gateways, resource consumers and producers, and the remote cloud platform are implemented as independent objects able to interact with the rest of the architecture, thus providing the main *Future-MCC* functionalities (i.e., data registration, resource discovery, data retrieving, etc.). Gateway devices also implement *Context Aware Processing* functionalities. During the simulation, data generated by a resource producer are stored in the gateway node for a given time interval. Without loss of generality, the lifetime of a cached content is set to the inverse of the corresponding request rate. Thus, when the gateway receives a request for a (not expired) cached content, it will reply to the resource consumer without contacting the resource producer. Otherwise, the request will be forwarded to the resource producer behind the gateway.

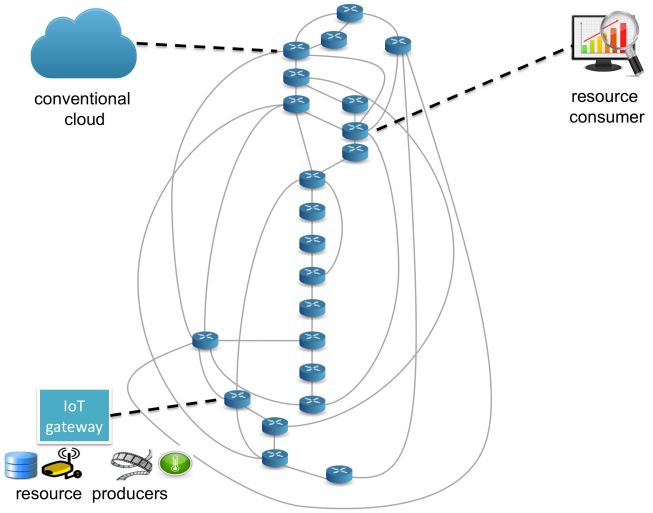


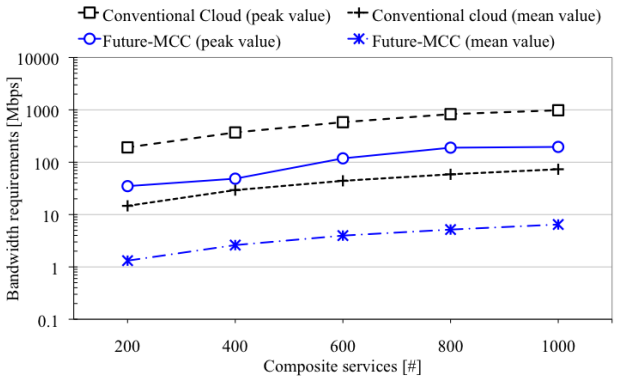
Fig. 5. The simulated GEANT Network, with an example configuration showing the position of the conventional cloud platform, a resource consumer, and a group of resource producers attached to a single gateway device.

We assume that resource producers generate real-time contents. In line with recent results discussed in the literature, we disabled any caching mechanism in the core network (the impact of the cache is irrelevant during the provisioning of real-time contents [29]).

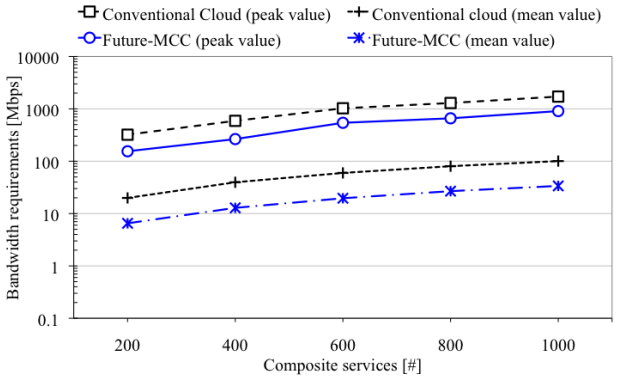
Finally, all the simulation results are averaged over 200 runs and both mean and peak values of the considered performance indexes (i.e., bandwidth requirement at the network edge, bandwidth requirement in the core network, and energy consumption of resource producers) are measured. This will provide a clear idea on the system demands related to both *Future-MCC* and the baseline solution.

C. Results

Focusing the attention on the access network, Figure 6 shows the bandwidth requirements as a function of the network load. As expected, this performance index increases with N_p and N_c. However, results clearly demonstrate that *Future-MCC* always ensures the lowest bandwidth demands. Thanks to its ability to discover and provide resources only when needed, in fact, it significantly reduces the amount of data that is exchanged at the access network. Differently, in a conventional cloud computing platform, all the resources are obliged to send data to a remote server (the cloud, for instance), even if only few of them are really transmitted back to the consumers. In general, this brings to a higher amount of data handled at the network edge. In addition, the access link connecting the cloud to the rest of the network is in charge to manage the highest traffic load (see peak value in Figure 6), thus becoming the bottleneck of the whole system. On the contrary, *Future-MCC* always registers the lowest peak bandwidth usage (note that this value is measured for the access links where the most popular resource producer is attached to).



(a)

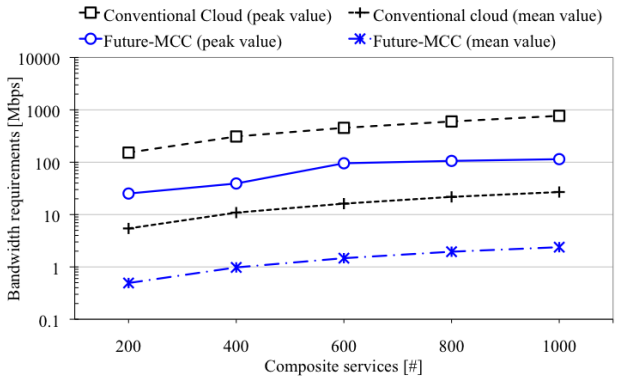


(b)

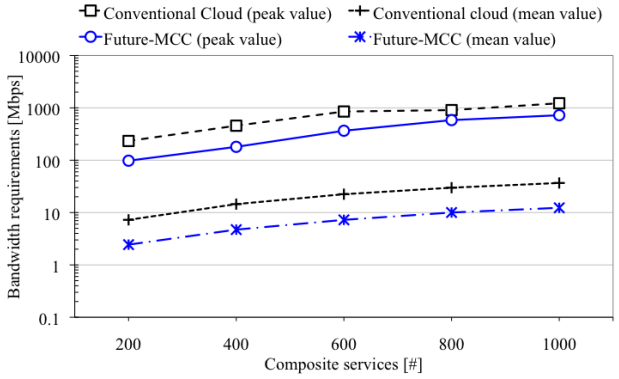
Fig. 6. Bandwidth requirements at the access network, when (a) $N_c = 0.1N_p$ and (b) $N_c = 0.5N_p$.

Similar considerations hold for the core network. According to results in Figure 7, *Future-MCC* always ensures the lowest bandwidth demands. In this case, in fact, the core network just delivers data that are actually requested by resource consumers. With the conventional cloud, the highest bandwidth usage is due to two reasons. First, requested data are twice transmitted (firstly from the producer to the cloud; then from the cloud to the consumer). Second, data that are not requested

are anyway delivered to the cloud, thus wasting a huge amount of bandwidth in the core network.



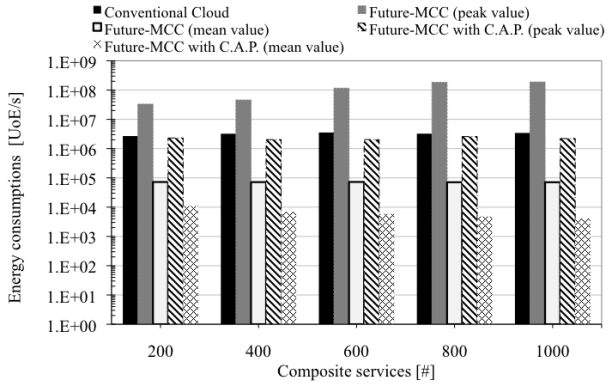
(a)



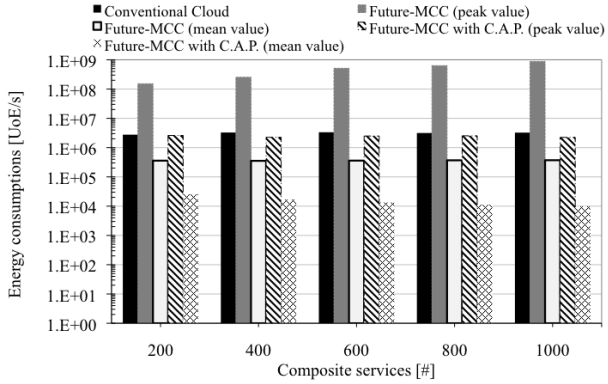
(b)

Fig. 7. Bandwidth requirements in the core network, when (a) $N_c = 0.1N_p$ and (b) $N_c = 0.5N_p$.

To provide a further insight, Figure 8 shows the energy consumption, expressed as the number of *unit of energy* that resource producers spend during the time. When the conventional cloud platform is used, energy consumption does not depend on the traffic load: each resource producer always sends the same amount of data to the remote cloud, thus wasting the same amount of energy. *Future-MCC* registers a different behavior: energy consumption is influenced by both traffic load and the presence of the *Context Aware Processing* functionalities. From Figure 8, it is possible to observe that the absence of the *Context Aware Processing* brings to the highest peak energy consumption. In this case, in fact, popular resource producers send more times the same data to the requesters, thus wasting their energy. When *Context Aware Processing* capabilities are enabled, instead, the peak energy consumption reaches lower values, comparable to those registered by the conventional cloud. In both cases, in fact, the most popular resource producer transmits the same amount of data during the same time interval. More in general, however, *Future-MCC* always ensures the lowest average energy consumption. Unlike the conventional cloud computing platform, in fact, resource producers do not send data if not requested by consumers, thus saving energy.



(a)



(b)

Fig. 8. Energy consumptions, when (a) $N_c = 0.1N_p$ and (b) $N_c = 0.5N_p$. C.A.P means *Context Aware Processing*

VII. Conclusions

In this paper we formulated a novel architecture, namely *Future-MCC*, able to efficiently support advanced applications in future Mobile Cloud Computing systems. By following the evolution of the current cloud computing paradigms toward the Mobile Cloud Computing and the Cloud of Things visions, we envisioned a futuristic scenario where a wide variety of devices will be able to leverage resources (embracing computational, storage, and sensing capabilities), which are dynamically offered in the local neighborhood and by remote data centers. Such a scenario will introduce a number of challenging issues and requirements (interoperability between different low-layer technologies and applications, heterogeneous and dynamic resources and network conditions, scalable resource discovery and access) that ask for innovative architectural and protocol design. To this end, we proposed an architecture that integrates some of the promising paradigms proposed

in the Future Internet research arena, that are middleware-based virtualization, Information Centric Networking, and SDN/NFV. In particular, they are jointly used and extended to provide effective and efficient mechanisms to *represent*, *discover*, and *access* resources. The architecture has been devised to be general-purpose and not with a specific service in mind. Indeed, its functionalities are described at a high level and no algorithms are specified in detail. Finally, the effectiveness of the conceived *Future-MCC*, as well as the performance gain offered with respect to conventional cloud computing platforms, have been investigated through computer simulations. Achieved results clearly demonstrate that the proposed approach ensures *(i)* a reduction of the bandwidth requirements spanning from 66% to 91% and *(ii)* an average energy saving equal to 99%. In the future, we plan to build a prototype to showcase the viability of the proposed approach in different pioneering use cases expected for upcoming MCC architectures.

ACKNOWLEDGMENTS

This work was performed in the context of the project BON-VOYAGE, which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 635867, and by the national research project PON03PE_00050_2 DOMUS "Home automation systems for a cooperative energy brokerage service".

REFERENCES

[1] M. Sadiku, S. Musa, and O. Momoh, "Cloud computing: Opportunities and challenges," *IEEE Potentials*, vol. 33, no. 1, pp. 34–36, Jan 2014.

[2] L. Heilig and S. Voss, "A scientometric analysis of cloud computing literature," *IEEE Trans. on Cloud Comput.*, vol. 2, no. 3, pp. 266–278, Jul. 2014.

[3] S. Sakr, A. Liu, D. Batista, and M. Alomari, "A survey of large scale data management approaches in cloud environments," *IEEE Surv. & Tuts.*, vol. 13, no. 3, pp. 311–336, Third 2011.

[4] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, pp. 1587–1611, 2013.

[5] A. Khan, M. Othman, S. Madani, and S. Khan, "A survey of mobile cloud computing application models," *IEEE Surv. & Tuts.*, vol. 16, no. 1, pp. 393–413, First 2014.

[6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. of ACM MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.

[7] E. Miluzzo, "I'm Cloud 2.0, and I'm Not Just a Data Center," *IEEE Internet Comput.*, vol. 18, no. 3, pp. 73–77, May 2014.

[8] N. Mitton, S. Papavassiliou, A. Puliafito, and K. S. Trivedi, "Combining cloud and sensors in a smart city environment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–10, 2012.

[9] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Surv. & Tuts.*, vol. 16, no. 1, pp. 414–454, First 2014.

[10] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of Information-centric Networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, 7 2012.

[11] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Surv. & Tuts.*, vol. 17, no. 1, pp. 27–51, 2015.

[12] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Surv. & Tuts.*, 2015.

[13] C.-A. Chen, M. Won, R. Stoleru, and G. G. Xie, "Energy-efficient fault-tolerant data storage & processing in mobile cloud," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 28–41, March 2014.

[14] Q. Han, S. Liang, and H. Zhang, "Mobile cloud sensing, big data, and 5g networks make an intelligent and smart world," *Network, IEEE*, vol. 29, no. 2, pp. 40–45, 2015.

[15] M. Gerla, "Vehicular cloud computing," in *Proc. of IEEE Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2012, pp. 152–155.

[16] M. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Surv. & Tuts.*, 2012.

[17] J. Zhou, T. Leppanen, E. Harjula, M. Ylianttila, T. Ojala, C. Yu, H. Jin, and L. T. Yang, "Cloudthings: A common architecture for integrating the internet of things with cloud computing," in *Proc. of IEEE Int. Cong. on Computer Supported Cooperative Work in Design (CSCWD)*, 2013, pp. 651–657.

[18] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 369–392, 2014.

[19] "Cisco forecasts, http://www.cisco.com/web/solutions/sp/vni/vni_forecast_highlights/index.html."

[20] M. Ouedraogo, S. Mignon, H. Cholez, S. Furnell, and E. Dubois, "Security transparency: the next frontier for security research in the cloud," *Journal of Cloud Computing*, vol. 4, no. 1, p. 12, 2015.

[21] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, Privacy & Trust in Internet of Things: the road ahead," *Computer Networks (Elsevier)*, vol. 76, pp. 146–164, 2015.

[22] M. Vučinić, B. Tourancheau, F. Rousseau, A. Duda, L. Damon, and R. Guizzetti, "Oscar: Object security architecture for the internet of things," *Ad Hoc Networks*, 2014.

[23] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," *IEEE Commun. Mag.*, vol. 49, no. 7, pp. 26–36, 2011.

[24] "ETSI TS 102 690 v2.1.1: Machine-to-Machine Communications (M2M); Functional Architecture," 2013.

[25] L. A. Grieco, M. B. Alaya, T. Monteil, and K. K. Drira, "Architecting Information Centric ETSI-M2M systems," in *Proc. of IEEE PerCom*, 2014.

[26] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, "Toward a standardized common M2M service layer platform: Introduction to oneM2M," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 20–26, June 2014.

[27] D. Matsubara, T. Egawa, N. Nishinaga, V. Kafle, M.-K. Shin, and A. Galis, "Toward future networks: A viewpoint from ITU-T," *IEEE Commun. Mag.*, vol. 51, no. 3, pp. 112–118, 2013.

[28] G. Xylomenos et al., "A Survey of Information-Centric Networking Research," *IEEE Surv. & Tuts.*, vol. PP, no. 99, pp. 1–26, 2013.

[29] G. Piro, L. A. Grieco, G. Boggia, and P. Chatzimisios, "Information-centric networking and multimedia services: present and future challenges," *ETT, Transactions on Emerging Telecommunications Technologies,*, vol. 25, no. 5, pp. 392–406, Apr. 2014, doi:10.1002/ett.2741.

[30] M. Bari et al., "A survey of naming and routing in information-centric networks," *IEEE Commun. Mag.*, vol. 50, no. 12, pp. 44–53, Dec. 2012.

[31] Y. Zhang, D. Raychadhuri, L. A. Grieco, E. Baccelli, J. Burke, R. Ravindran, G. Wang, A. Lindren, B. Ahlgren, and O. Schelen, "Requirements and Challenges for IoT over ICN, IRTF Internet Draft, draft-zhang-icnrg-icniot-requirements-00," IRTF, Internet Draft, Nov. 2015.

[32] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, "Information-centric networking for the internet of things: challenges and opportunities," *IEEE Network*, vol. 30, no. 2, pp. 92–100, Mar. 2016.

[33] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named Data Networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[34] D. Smetters and V. Jacobson, "Securing Network Content," Tech. Rep., October 2009.

[35] Y. Zhang, D. Raychadhuri, L. A. Grieco, S. Sabrina, H. Liu, and G. Wang, "ICN based Architecture for IoT, IRTF Internet Draft, draft-zhang-icn-iot-architecture-01," IRTF, Internet Draft, Mar. 2016.

[36] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *IEEE Surv. & Tuts.*, vol. 16, no. 4, pp. 2181–2206, Fourth quarter 2014.

[37] C. Yi *et al.*, "On the role of routing in Named Data Networking," Technical report, Named-Data Networking Project, Tech. Rep., 2013.

[38] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, and G. Ferrari, "IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios," *Sensors Journal, IEEE*, vol. 15, no. 2, pp. 1224–1234, Feb 2015.

[39] S. L. Keoh, S. Kumar, and H. Tschofenig, "Securing the Internet of Things: A Standardization Perspective," *IEEE IoT J.*, vol. 1, no. 3, pp. 265–275, Jun. 2014.
[40] S. Sciancalepore, A. Capossele, G. Piro, G. Boggia, and G. Bianchi, "Key management protocol with implicit certificates for iot systems," in *Proc. of ACM Int. Workshop on IoT challenges in Mobile and Industrial Syst. (IoT-Sys)*, Florence, IT, May 2015.
[41] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 1999, pp. 126–134.
[42] "Geant project website, http://www.geant.net/."

Giuseppe Piro (S'10-M'13) is an Assistant Professor at "Politecnico di Bari", Italy. He received a first level degree and a second level degree (both cum laude) in Telecommunications Engineering from "Politecnico di Bari", Italy, in 2006 and 2008, respectively. He received the Ph.D. degree in Electronic Engineering from "Politecnico di Bari", Italy, on March 2012. His main research interests include quality of service in wireless networks, network simulation tools, 4G and 5G cellular systems, Information Centric Networking, nano communications, and Internet of Things. He founded both LTE-Sim and NANO-SIM projects and is a developer of Network Simulator 3. Actually, he is also participating to standardization activities in IETF 6tisch and IEEE P1906.1 working groups.

Marica Amadeo is a Postdoc researcher at University Mediterranea of Reggio Calabria, Italy. She received a Bs Degree (2005) and a Ms Degree (2008) in Telecommunications Engineering from the University Mediterranea of Reggio Calabria, and a Ph.D. degree in 2013 from the same University. Her major research interests are in the field of information-centric networking and wireless ad hoc networks. She authored more than 30 international publications including journals, conferences and book chapters. She served as a reviewer for several international conferences and journals. She also received a best reviewer award by IEEE Communications Letters in 2014 and 2015 and a Best Paper Award nomination at NOF 2011.

Gennaro Boggia (S'99-M'01-SM'09) received, with honors, the Dr. Eng. Degree in Electronics Engineering in July 1997 and the Ph.D. degree in Electronics Engineering in March 2001, both from the "Politecnico di Bari", Italy. Since September 2002, he has been with the Department of Electrical and Information Engineering at the "Politecnico di Bari", Italy, where he is currently Associate Professor. From May 1999 to December 1999, he was visiting researcher at the "TILab", TelecomItalia Lab,Italy, where he was involved in the study of the Core Network for the evolution of 3G cellular systems. In 2007, he was visiting researcher at FTW (Vienna), where he was involved in activities on passive and active traffic monitoring in 3G networks. He has authored or co-authored more than 100 papers in international journals or conference proceedings. His research interests span the fields of Wireless Networking, Cellular Communication,Information Centric Networking, Internet of Things (IoT),Protocol stacks for industrial applications and smart grids, Internet measurements, Network Performance Evaluation. Currently, he serves as Associate Editor for the Springer Wireless Networks journal.

Claudia Campolo is an Assistant Professor of Telecommunications at University Mediterranea of Reggio Calabria, Italy. Before her current appointment, she received a Laurea degree in Telecommunications Engineering (Oct. 2007) and a PhD degree (Feb. 2011) from the University Mediterranea of Reggio Calabria, Italy. Since March 2011 she has been with the same university as a Post-Doc researcher. In 2008 she was a visiting PhD student at the Department of Electronics Engineering of Politecnico di Torino and a DAAD fellow at University of Paderborn, Germany (March 2015-April 2015). Her main research interests are in the field of wireless and vehicular networking and future Internet architectures. She authored more than 60 international publications including journals, conferences and book chapters. She has received three Best Paper Awards for research into vehicular networks and given tutorials at international conferences on this topic. She is co-editor of the book "Vehicular ad hoc networks: standards, solutions and research", published by Springer-Verlag in June 2015.

Luigi Alfredo Grieco (S'02-M'04-SM'12) is an Associate Professor in Telecommunications at "Politecnico di Bari". Formerly he has been Visiting Researcher with INRIA (Sophia Antipolis, France) in 2009 and with LAAS-CNRS (Toulouse, France) in 2013, working on Internet measurements and M2M systems, respectively. He has authored more than 100 scientific papers published in international journals and conference proceedings of great renown that gained more than 1000 citations. His main research interests include TCP congestion control, quality of service in wireless networks, IoT, and Future Internet. He serves as editor of the IEEE Transactions on Vehicular Technology (for which he has been awarded as top associate editor in 2012) and as Executive Editor of the Transactions on Emerging Telecommunications Technologies (Wiley). Within the IETF and IRTF, he is actively contributing to the definition of new standard protocols for industrial IoT applications and new standard architectures for tomorrow ICN-IoT systems.

Antonella Molinaro is an associate professor of Telecommunications at the University Mediterranea of Reggio Calabria, Italy. Before, she was with the University of Messina (1998-2001) and the University of Calabria (2001-2004) as an assistant professor; with the Polytechnic of Milano as a research fellow (1997-1998); and with Siemens A.G., Munich, Germany as a CEC fellow in the RACE-II program (1994-1995). She graduated in Computer Engineering (1991) at the University of Calabria, received a Master diploma in Information Technology from CEFRIEL/Polytechnic of Milano (1992), and a Ph.D. degree in Multimedia Technologies and Communications Systems (1996). Her current research activity focuses on wireless networking, vehicular networks, information-centric networking. She participates in the Information Centric Networking research group (ICNRG) of IRTF and is a member of the NetWorld2020 European Technology Platform. She is in the Editorial board of Computer Networks; Transactions on Emerging Telecommunications Technologies; International J. of Distributed Sensor Networks; EAI Transactions on Internet of Things.

Giuseppe Ruggeri received the MS degree in Electronics Engineering in 1998 from the University of Catania (Italy). In 2002 he received the Ph.D. in Electronics, Computer Science and Telecommunications engineering from the University of Palermo (Italy). Since 2002 he has been Assistant Professor in Telecommunications at the University "Mediterranea" of Reggio Calabria (Italy). He authored more than 50 international publications including journals, conferences and book chapters on wireless and mobile systems, protocols, modeling and simulation. He served as TPC member of several leading international conferences (ICC, GLOBECOM, WCNC and others). He served as general co-chair of the SWANSITY 2014 and SWANSITY 2015 workshops. He joined several research projects financed either by Calabria local Government, Italian National Government or European Union. He served as the coordinator of the local research teams at University "Mediterranea" involved in the projects "Stem"-Net devices to build a Self-Generating access network and DOMUS both financed by the Italian Government. His current interests include the interconnection-integration of heterogeneous wireless networks, self-organizing networks, and M2M communication.