Supporting the Development of Next-generation Fog Services

Carlo Giannelli
University of Ferrara, Ferrara, Italy
Dep. of Mathematics and Computer Science
carlo.giannelli@unife.it

Filippo Poltronieri, Cesare Stefanelli, Mauro Tortonesi
University of Ferrara, Ferrara, Italy
Department of Engineering
{filippo.poltronieri, cesare.stefanelli, mauro.tortonesi}@unife.it

Abstract—Fog Computing is a recent and compelling paradigm that proposes to run information-processing services at the edge of the network. While interesting standardization efforts are being currently pursued by many organizations, most of them focus on management and orchestration functions and primarily propose the adoption of programming models designed for Cloud applications in the Fog. Instead, Fog Computing applications would significantly benefit from innovative solutions that adopt an “acceptable lossiness” perspective and manage information processing and dissemination in a dynamic and integrated fashion. This paper presents an overview of the opportunities and challenges in Fog Computing application development, proposes an innovative holistic Software Defined Networking (SDN) approach based on an information-centric and value-based service model to support those applications, and presents an overview of the Holistic Analytics and Networking (HAN) SDN architecture that we are developing to realize this ambitious vision.


I. INTRODUCTION

Fog Computing is a new paradigm that proposes to run information-processing services at the edge of the network, that is, on top of edge devices in proximity of either raw data sources, information consumers, or both. This represents a very interesting model, designed to integrate with the centralized Cloud-based computation paradigm, significantly extending its capabilities. In fact, the concept of Cloud-things continuum recently emerged as a (quasi-)continuous segment that connects the Cloud with Internet of Things (IoT) devices, along which service instances processing IoT data can be instantiated [1].

Several standards are emerging for Fog Computing applications [2]. ETSI oneM2M is a high-level architecture defining how application and common components should interact in a Fog platform. The ETSI Multi-access Edge Computing (MEC) technical group proposed a reference architecture for orchestration of Fog applications. The ETSI Network Functions Virtualization Management and Orchestration (NFV MANO) technical committee proposed another orchestration architecture. The OpenFog Consortium started working towards a reference architecture as well. Others proposed to reuse architectural and service models, e.g., based on containerization technologies, microservices, or edge devices [3].

However, most of these efforts focus almost exclusively on orchestration and assume the adoption of programming and service models and communication protocols designed for Web / Cloud applications or their corresponding scaled-down versions, such as Constrained Application Protocol (CoAP) [4], designed for constrained devices. Instead, we believe that Fog Computing applications would significantly benefit from holistic approaches that leverage adaptive, value-based, and information-centric service architectures and adopt innovative Software Defined Networking (SDN) solutions to dynamically control information dissemination.

More specifically, Fog services should adopt an “acceptable lossiness” perspective and attempt to achieve the desired Quality of Experience (QoE) levels by focusing on the processing and dissemination of the most valuable pieces of information from the end users’ point of view. In addition, service components should be possibly instantiated along the information communication path and their processing logic should be adaptive and capable of reducing its resource requirements to match the ones available in their current instantiation location. Finally, the specific dissemination solution for processed information should be tailored according to the service requirements and the current execution context, dynamically choosing the most suitable solution among a wide array of different options.

These approaches would naturally enable a much more effective use of the scarce and heterogeneous computation and bandwidth resources in information processing and dissemination, and facilitate service composition, reuse, and redeployment, particularly important in dynamic scenarios such as forthcoming Smart Cities. This would significantly help the development of applications capable of taking full advantage of Fog Computing while addressing the most important challenges proposed by this new environment.

The paper presents an overview of the opportunities and challenges in Fog Computing application development in Smart Cities (Section II), proposes an innovative holistic SDN approach based on an information-centric and value-based service model to support those applications (Section III), and presents an overview of the Holistic Analytics and Networking (HAN) SDN architecture that we are developing to realize this ambitious vision (Section IV).
II. FOG COMPUTING IN SMART CITIES

Smart Cities are probably the most interesting application scenario for Fog Computing. Their networking infrastructure is characterized by an edge, in which a plethora of smart objects, sensors, vehicles, and personal devices provide capillary sensing functions, and by a connected core, where the collected raw data is stored and processed through sophisticated analytics in metropolitan areas and Cloud level data centers.

From the communication perspective, there are two main types of edge networks: IoT networks, that connect IoT devices one each other, and end user networks, that connect users with the rest of the network using wide range wireless communication protocols such as WiFi and LTE/4G.

Typically, IoT devices are deployed in groups to form sophisticated sensing systems and communicate based on IoT networks exploiting a multitude of short-range and low-power wireless communications protocols, e.g., IEEE 802.15.4, Bluetooth LE, and NFC. Then, IoT networks are connected to other wider networks (or even the Internet) through one or more IoT gateway devices. IoT gateways typically have significantly better computation, storage, and communication capabilities if compared with IoT devices.

End user networks are typically heterogeneous networks formed by a set of overlapping medium-range wireless networks of different types, including cellular networks with macro and pico cells, Wi-Fi networks, etc. Device-to-device communications, such as WiFi-direct and LTE-direct, further expand communication opportunities by allowing personal devices to share information of common interest when they are in proximity.

Fog Computing extends this scenario by allowing IT service developers and providers to allocate (a portion of the) information-processing tasks at the edge of the network, with the potential of significantly reducing the response latencies (and consequently improving the quality) of IT services and reducing the burden on the network infrastructure. As a result, Fog Computing represents a particularly attractive paradigm for the development of low latency, deeply immersive, and high-value-added IT services designed for digital citizens.

In this scenario, information processing tasks can be allocated either in the Cloud or on top of a plethora of different edge devices, including the aforementioned IoT gateways, Cloudlets or Micro-Clouds, and Multi-Access Edge Computing, [1]. Also note that users’ smart personal devices can, and typically do, perform a dual role: they access Fog Services and at the same time they operate as raw data sources thanks to the recent and remarkable developments in their sensing capabilities.

However, the implementation of Fog Computing solutions in Smart City environments represents a very difficult task for three main reasons. First and foremost, processing a massive amount of raw data generated by the IoT (predicted to increase to as high as 850 ZB per year by 2021 [5]) with the scarce computational and energy resources available to edge devices represents a daunting challenge. In addition, the significant mobility of users and terminals [6] causes frequent communication disruptions and wide variability in channel performance. Finally, edge resources are very dynamic and heterogeneous.

III. A HOLISTIC SDN-BASED APPROACH FOR FOG NETWORKING

Fog Computing applications would significantly benefit from innovative solutions that adopt an “acceptable lossiness” perspective and a holistic approach that integrate value-based and information-centric analytics and dissemination solutions. In particular, we propose a novel SDN-based application design paradigm that considers two most relevant aspects. We deem as of paramount importance, on the one hand, the capability of adopting value-based information and service models, that can automatically skip low-value-added raw data to focus on timely and reliable processing and dissemination of high-value-added one. On the other hand, advanced Fog-specific SDN approaches, that can map edge communications over different networking protocols and information dissemination solutions according to the current context.

A. The AIV Service Model

Fog services mostly involve continuously running information processing tasks. This has been recognized by researchers, who have proposed information-centric programming and service models for Fog applications, such as the Distributed Dataflow that is inspired to digital signal processing and allows BPEL-like service definition through scripting solutions [7]. Those programming models mostly focus on the realization of services through the orchestration of functions realized by specific components, without explicitly addressing how to implement potentially expensive information processing tasks on resource-scarce Fog environments.

Instead, the resource scarcity and dynamicity of Fog environments suggests the opportunity to push forward the state of the art and experiment with lossy and adaptive service models. Such solutions should be capable of automatically scaling their resource requirements to their current execution context, by also dynamically evaluating the value of dispatched information in traffic flows from an application point of view to, e.g., appropriately tune their allocated bandwidth in relation
to the importance of carried information and the network current state.

More specifically, a particularly interesting approach would be to explicitly consider the different characteristics of Information Objects (IOs) to prioritize the processing and dissemination of the most important IOs, while discarding low value ones [8]. To this end, we developed the Adaptive, Information-centric, and Value-based (AIV) model for Fog services.

First of all, AIV explicitly considers an information maturity model designed for Fog applications. It defines IOs of 3 different maturity classes: raw data IOs collected from the IoT and/or the end-user layer, distilled IOs produced from a first level analysis of raw data through some basic aggregation and/or processing functions, and Consumption-Ready IOs (or CRIOs) that contain fully-processed information that can be forwarded to users in response to their service requests. Within each class, IOs can have different maturity levels to allow, e.g., the combined use of several processing components to produce distilled IOs from raw data IOs (or CRIOs from distilled IOs).

On top of this information maturity model, AIV adopts a modular service architecture that maximizes the opportunities for reuse of processing components and of generated results. More specifically, a Fog service is implemented through the coordinated efforts of (at least) two different processing layers: pipelines and services. Pipelines produce distilled IOs by analyzing raw data IOs (or distilled IOs with lower maturity levels) and are typically designed to provide basic functions, e.g., OCR processing or face detection in images. Services generate CRIOs by further analyzing distilled IOs (or CRIOs with lower maturity levels) through application-specific processing. Both pipeline and service components can be moved from the Cloud to the Fog and vice versa according to current Fog service requirements and execution context.

Explicitly tracking the maturity of IOs as they undergo several processing stages facilitates service composition and helps to make effective decisions on which IOs to prioritize for processing and dissemination. In this context, our experience suggests that Quality-of-Information (QoI) and Value-of-Information (VoI) represent particularly attractive metrics for IO prioritization [9]. QoI represents a static and objective metric that considers the intrinsic characteristics of an IO and typically depends from the type of IO considered. For instance, the QoI of a digital sound recording could be defined by the sample size, the sample rate, etc. VoI instead represents a dynamic and subjective metric that classifies an IO according to the utility it provides to its consumer. For instance, IOs whose content does not add significant value to the knowledge already built from the analysis of previous IOs of the same type would have a low VoI.

Classifying IOs according to their intrinsic QoI and the VoI they provide to their recipients represents a very natural and effective criterion for prioritizing the most important data for processing and dissemination. In the likely case that the shortage of computation, energy, or bandwidth resources in the Fog layer will not support the processing and dissemination of all IOs received, IOs with the lowest associated QoI and/or VoI could be dropped. This approach would enable to maximize in a natural and straightforward fashion the utilization of Fog resources from the user perspective, also addressing the issue of resource sharing among competing Fog services.

B. Extending SDN to Fog Computing Scenarios

While the SDN approach has been traditionally adopted (and by now it is considered the standard solution) in data centers and huge enterprise networks [10], some efforts demonstrate its validity also when adopted in Fog Computing [6], [11]–[13]. However, its adoption on the edge side of Fog environments needs more work since there is the need to adopt a wider and holistic point of view to support Fog service reconfiguration not only considering networking features and resources, but also service composition based on IOs/CRIOs and tuning exploiting application-level information such as QoI and VoI.

In particular, as illustrated in Fig. 1, we propose the adoption of a Fog-specific SDN solution based on programmable routers capable of hosting information processing tasks and a SDN Controller capable of instantiating information processing and dissemination tasks in an integrated and context-aware fashion. More specifically, the SDN Controller is in charge of, on the one hand, managing edge devices to tune their behavior in relation to current active Fog services and, on the other hand, interacting with Cloud and mobile network SDN Controllers to leverage the full exploitation of available resources in other domains to maximize the QoE.

Delving into finer details about edge-specific features of the Fog SDN routers, we propose a two-layer approach, a top service layer and a bottom communication layer. In both layers edge devices going to generate traffic interact with the Fog SDN Controller to provide their own application requirements, e.g., specifying that they are going to send a multimedia stream to one or more destination nodes with a given average bitrate for a given time period. The Fog SDN Controller exploits its global knowledge of the edge topology and its current state to appropriately reconfigure the edge network and its nodes to both dynamically compose services and tune service provisioning performance.

At the service layer the Fog SDN Controller dynamically deploys and de/activates application-specific components on edge devices. To this purpose, the SDN Controller exploits received application requirements to also identify required processing and filtering features intermediate nodes should support. Then, it dynamically deploys (eventually) missing software components to support the effective provision of required fog services. At the communication layer the Fog SDN Controller dynamically manages edge devices to improve the QoS of packet dispatching by computing best paths towards the destination and tuning forwarding rules on intermediate nodes.

To this purpose, we propose to adopt three mechanisms: pure IP forwarding, (Vol-oblivious) overlay-based dissemination, and VoI-aware overlay-based dissemination.
**Pure IP forwarding** works by dynamically modifying per-device routing tables with traditional iptables command to reroute vanilla UDP and TCP traffic flows based on IP address destination. This mechanism allows to manage legacy applications in a completely transparent fashion from edge device point of view, thus making easy the QoS management.

**Overlay-based dissemination** supports packet dispatching among edge devices in a collaborative fashion by adopting routing schemes based on flow ids. In this case, the Fog SDN Controller provides a flow id to applications to label generated traffic with. In addition, the Fog SDN Controller manages nodes of the Fog environment specifying, e.g., that the traffic labeled with a given flow id has higher priority and thus other lower priority traffic flows should be temporarily delayed.

**VoI-aware overlay-based forwarding** extends the previous mechanism allowing to tune packet forwarding not only based on the flow id, but also based on the value of information carried by each packet. To this purpose, the SDN Controller manages edge nodes also providing routing rules based on QoI and VoI values. Let us note that in this manner it is possible to support QoI/VoI dependent information dissemination, e.g., by specifying that packets carrying data with QoI/VoI values below a threshold should be discarded or also differentiating destination nodes on QoI/VoI ranges not only at the sender, but also at intermediary nodes. In other words, this mechanism allows to dynamically modify the pipeline of fog services also specifying different pipelines for the same data based on QoI/VoI values.

It is worth noting that the Fog SDN Controller can selectively adopt one or more of above mechanisms with a per-application granularity, depending on the current state of the network and specific needs of applications. For instance, the Fog SDN Controller can decide that a simple application but with strict low latency requirements should exploit pure IP forwarding (with no overhead) while more articulated applications can be effectively provided only with service layer supported by additional software component deployed dynamically coupled with value-dependent packet dissemination. In addition, above mechanisms can be adopted at both service and communication layers. In the former case, flow id and QoI/VoI information can be exploited to reroute traffic flows to different nodes (also eventually adopting tree-based multicasting techniques) to dispatch packets to the node interested to them. In the latter case, the same information can be exploited to tune the QoS of traffic flows, e.g., by slightly delaying (or even temporarily discarding) traffic flows with lower assigned priority level or with lower QoI/VoI values to ensure additional network resources to other more important traffic flows.

IV. THE HAN SDN SOLUTION

We devised the Holistic Analytics and Networking (HAN) SDN-based solution to implement the model discussed in the previous Section. HAN integrates and extends our past work on the Real Ad-hoc Multi-hop Peer-to-peer (RAMP) middleware [14] with SDN capabilities to support end-to-end QoS [15], [16] and the Sieve, Process, and Forward (SPF) middleware [8], supporting the creation of overlay layers to enable the dynamic deployment composition of Fog services.

Fig. 2 depicts the overall architecture of a Fog service node in HAN. The components presented in the figure are deployed and activated on each Fog service node to allow their participation in the SDN-enabled Fog environment. In addition, for each Fog environment there is one node acting as Fog SDN Controller by registering itself to the local RAMP as ”Fog SDN Controller” service, thus allowing remote Fog nodes to dynamically discover and register to it. Based on information provided by remote Fog nodes, the Fog SDN Controller generates a weighted graph. To this purpose, the current implementation of the Fog SDN Controller adopts the Graph Stream library\(^1\) allowing to apply, e.g., the Breadth First and the Dijkstra’s algorithm based on different cost functions to dynamically compute best paths to provide Fog services. For instance, when an application requires to the Fog SDN Controller the best path to access a service, it can specify one of the three already developed policies: IP-based, i.e., managing the Fog environment to access the remote service by adopting the traditional IP forwarding mechanism; overlay dispatching, i.e., exploiting the RAMP middleware to forward packets in a spontaneous fashion and dispatching packets based on a flow id, senders have to tag transmitted packets with; and VoI-based, identifying the path towards the destination based on the dynamically calculated VoI of packets, e.g., to exploit large bandwidth and small latency for packets with high VoI (thus ensuring better QoE) and less capable paths for packets with low VoI.

Delving into finer details of each Fog node, the overall architecture is divided into the Control Plane and the Data Plane. The Control Plane (Fig. 2, top) primarily consists of Link Connectivity Manager (LCM) and Control Agent (CA). LCM manages single-hop links and provides network status information by exploiting the OSHI library\(^2\) on desktop OSs and TrafficStats on Android devices. CA gathers information, sends it to the controller, and receives commands from the controller. More specifically, the Communication sub-component appropriately controls the underlying mechanisms to dispatch packets by dynamically de/activating the currently enforced management policy with a per flow granularity, while the Service sub-component dynamically deploys and activates software modules to enrich Fog nodes with additional capabilities required to correctly provide requested services.

The Data Plane (Fig. 2, bottom) is composed of the RAMP middleware, the general-purpose Data Plane Forwarder (DPF) component specializations, and the SPF middleware enhanced for Fog service dynamic composition. The RAMP middleware supports the creation of the multi-hop overlay network with best-effort packet dispatching. DPF properly manages packets received by the RAMP middleware on a per-flow basis, by adopting a listener approach that allows to intercept any

\(^1\)https://github.com/oshi/oshi
\(^2\)http://graphstream-project.org/
incoming packet. In addition, DPF delegates packet management to one or more service modules managed by SPF, in relation to the information provided to SPF by the Service sub-component.

Note that the development of QoI- and VoI-aware Fog services typically requires explicit support at the middleware level. In fact, there is the need to have QoI and VoI evaluation systems, that are service dependent. VoI estimation is particularly challenging as the VoI of an IO will likely change not only if we consider a different recipient but also as the recipients’ context changes.

Our SPF middleware allows to define QoI-based content filters at the pipeline level [8]. First of all, pipelines can define static filters for dropping IO with an associated QoI that does not fall in the range allowed by service designers. In addition, pipelines can leverage differential and content-based filters that compare the content of new IOs to the ones of recently processed IOs and restrict the processing to only data containing significant amount of novel information. In SPF, services of each information processing component define VoI calculation policies. VoI is then used to prioritize information processing and communication.

Finally, the resource scarcity at the Fog layer suggests to consider different execution models for Fog services. For instance, in addition to “resident” (i.e., long-time running) Fog services, SPF enables to deploy “normally-off” Fog services, that are activated only on demand and for a short amount of time, and are automatically deactivated when not needed [17].

V. RELATED WORKS

Research efforts on Fog Computing can be divided into three main different visions, depending on how the Fog is perceived: an extension of the Cloud, a quasi-autonomous ecosystem, and a Cloud cooperative environment. In the first vision, Fog Computing could be seen as an extension of the Cloud that enables the realization of highly-responsive Cloud services at the edge. The idea is to exploit the data sources and consumer nodes proximity [18] [19] to reduce bandwidth utilization, service interruptions due to temporary Cloud outages, latency, and processing time of tasks that could be executed at the edge [20].

This first approach is usually based on virtualization techniques to orchestrate and manage the deployment of applications and resources. One example is the one provided in [21], where a chemistry inspired osmosis approach is described to simplify the migration of applications from the Cloud to the edge. The described approach makes use of platform agnostic container-based virtualization technologies to run micro-services on both Cloud and edge datacenters and SDN and network function virtualization (NFV) to provide network management abstraction functionalities. A similar approach can be found in [22], where a SDN architecture and a docker containerization technique are used to build a prototype for application management at the edge.

Instead, from an IoT-centric perspective, Fog Computing can be perceived as a service platform, independent or disconnected from the Cloud. In this case, the Fog orchestration can be considered as a distributed system that provides computational power and resources at the edge without necessarily relaying on the Cloud [23]. In this area, researchers focus on the efficient and dynamic deploying of resources and computation [24] among Fog nodes to ensure QoS [25] requirements at Fog consumers.

Finally, the last branch of research on Fog Computing considers the Fog and Cloud as coexisting ecosystems that can mutually benefit from one each other [26], [27]. Mutual benefits include offloading of computational intensive tasks, historical storage on the Cloud, and support for low-latencies services at the edge. Solutions in this area of research include smart gateways based architectures to support the Fog orchestration [28] and applications mobility between different stations [6]. The mutual collaboration between the Cloud and Fog pushes for solutions and innovative tools capable of optimally addressing task and resource allocation between the
two ecosystems also considering computational prices, energy consumption, and latency optimization [29]. Cloud-centric approaches, which aim at reusing at the edge the same platforms and technologies available on the Cloud could suffer from performance and coordination issues in severely constrained and distributed environments such as the ones where Fog Computing infrastructures are deployed. However, IoT-based approaches that mainly consider Fog Computing as a service, often independent from the Cloud, tend to present fragmentation in the view of resources to the applications and poor performances in the processing of intense computational tasks, when support from the Cloud is missing.

Our HAN SDN proposal finds its allocation in the third class of solutions, which tries to exploit both Cloud and Fog ecosystems and solve multiple research challenges in this area by proposing an innovative platform for distributing computation tasks between the Cloud and the Fog resources depending on their computational load, supporting service and resource discovery, and defining a common paradigm for Fog-enabled application development.

VI. CONCLUSIONS

The development of Fog Computing applications will significantly benefit from innovative solutions designed to prioritize the processing of the most valuable portion of information and to disseminate the results in a context-aware fashion. To this end, we devised the HAN SDN solution, which adopts an “acceptable loss” perspective for QoE provisioning and enables the effective use of the scarce and heterogeneous computation and bandwidth resources in Fog Computing environments, significantly facilitating the development of applications while addressing the most important challenges proposed by this new environment.

REFERENCES