

Article

Fog over Virtualized IoT: New Opportunity for Context-Aware Networked Applications and a Case Study

Paola G. V. Naranjo ¹ , Zahra Pooranian ² , Shahaboddin Shamshirband ^{3,4,*} ,
Jemal H. Abawajy ⁵  and Mauro Conti ² 

¹ Department of Information Engineering, Electronics, and Telecommunication, Sapienza University of Rome, via Eudossiana 18, 00184 Rome, Italy; paola.vinueza@uniroma1.it

² Department of Mathematics, University of Padua, 35131 Padua, Italy; zahra@math.unipd.it (Z.P.); conti@math.unipd.it (M.C.)

³ Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam

⁴ Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam

⁵ Faculty of Science, Engineering and Built Environment, Deakin University, Melbourne 3220, Australia; jemal.abawajy@deakin.edu.au

* Correspondence: shahaboddin.shamshirband@tdt.edu.vn

Academic Editor: Giancarlo Fortino

Received: 19 November 2017; Accepted: 13 December 2017; Published: 19 December 2017

Abstract: In this paper, we discuss the most significant application opportunities and outline the challenges in real-time and energy-efficient management of the distributed resources available in mobile devices and at the Internet-to-Data Center. We also present an energy-efficient adaptive scheduler for Vehicular Fog Computing (VFC) that operates at the edge of a vehicular network, connected to the served Vehicular Clients (VCs) through an Infrastructure-to-Vehicular (I2V) over multiple Foglets (FLs). The scheduler optimizes the energy by leveraging the heterogeneity of FLs, where the FL provider shapes the system workload by maximizing the task admission rate over data transfer and computation. The presented scheduling algorithm demonstrates that the resulting adaptive scheduler allows scalable and distributed implementation.

Keywords: Vehicular Fog Computing; Big Data Streaming; energy-efficiency; resource management; Infrastructure-to-Vehicular

1. Introduction

Vehicular Fog Computing (VFC) is a promising, as yet unexplored paradigm which aims to support services in order to include Internet infotainment applications. Cloud Computing (CC) has attracted big companies (such as VTube, Netflix, Cameleon, etc.) [1,2] that seek to exploit the Fog Nodes (FNs) of the Infrastructure-to-Vehicle (I2V) with single-hop IEEE 802.11 wireless links for data dissemination, providing low energy consumption [2].

Furthermore, Fog Computing (FC) is used to design an efficient solution supporting low-latency [3], geo-distributed Internet of things (IoT) devices/sensors [4], as well as providing energy-efficiency and self-adaption. It provides spectrum efficiency by means of a strict collaboration among end-devices in the same integrated platform [5]. FC aims at distributing the self-powered data center (e.g., FNs) between resource-remote clouds and a resource-limited smartphone in the vehicles [6], in order to perform context-aware energy-efficient data mining and dissemination for the adoption of a smartphone device and clients' highly dependent mobile applications [3,7,8]. Moreover, being deployed in the proximity of the Vehicular Clients (VCs), the FNs efficiently exploit the awareness of

the connection states supporting latency and delay-jitter. Among the contributions that deal with the adaptive scaling of the server resources in similar VCs, the work in [9] focuses on the impact of the dynamic scaling of the CPU computing frequencies on the energy consumption experienced by the execution of MapReduce-type jobs.

Fog Data Centers (FDCs) are dedicated to supervising the transmission, distribution and communication networks [10], all of which require a precisely managed grid infrastructure from generation points to consumption points by using the communication of measured values and transmitted control information more accurately. The IoT applications (real-time requirements, such as stream processing) are distributed in different geographical locations [11], with numerous devices and emit data via gateways (FNs) for further processing and filtering. In addition, the FNs host application modules that connect sensors to the Internet. FNs include Cloud resources that are provisioned on-demand from a geographically distributed Fog Data Center (FDC). Fog Data Center services are mainly implemented in Fog; they use a distributed architecture over moderate-bandwidth to provide high availability operations. The vehicles are attached to one of the Road Side Units (RSU) in the FN [12]; the heterogeneous type of devices consider a physical infrastructure, with each FN consisting of a single server or a set of servers building heterogeneous FNs.

Fog Data Centers entails a huge amount of computations and it is distributed; it may be more energy-efficient than the centralized Cloud model of computation, with the reduction of the FDC energy consumption being an important challenge. Also, this drastically reduces the traffic sent to the Cloud by allowing the placement of filtering operators close to the data sources. FDC, as a vital component of the IoT (vehicular) environment, is capable of filtering and processing a considerable amount of incoming data on edge devices, by making the data processing architecture distributable, thus scalable. Hence, it is an important task to provide a neat simulated scenario or case study, thus detailing the analytic structure of the FDC and the traffic injected to the engaged servers to make the presented model more efficient and interesting.

Fog Data Centers, in each processing unit, executes the currently assigned task by self-managing its own local virtualized storage/computing resources. When a request for a new job (transferred through the data network) is submitted from the remote clients and transferred through the Internet to the FNs and FDC, the resource controller dynamically performs both admission control and allocation of the available virtual resources.

Fog Data Centers, roughly speaking, is composed of (i) an Access Control Server and Router (ACSRs or Adaptive load dispatcher); (ii) a reconfigurable computing Cloud managed by the Virtual Machine Manager (VMM), called a load balancer, the related switched Virtual LAN; and (iii) an adaptive controller that dynamically manages all the available computing-communication resources, performs the admission control of the input/output traffic flows out to the ACSR and delivers the processed information to the FNs. As we know, many efforts have been made to curtail the energy consumption in data centers. Roughly speaking, FDC consolidation is a popular strategy to further reduce the energy consumption by turning OFF the underutilized VMs and grouping the VMs onto the smallest number of physical servers. The effectiveness of FDC consolidation in reducing the costs of IT is shown by the popularity of this strategy. The recent consolidation technologies employed in data centers encompass server and storage virtualization, as well as the deployment of tools for process automation. In this case study, we use the server virtualization as a dynamic control to improve the energy efficiency in FDC.

The Main Contribution of the Paper

The technical contribution of this paper focuses on the design of a Fog-based VFC architecture for the energy-efficient joint management of the networking and computing resources under hard constraints to improve the overall tolerated computing-plus-communication latency.

The main contributions of this paper are as follows:

- (i) To the best of our knowledge, there is no related work that considers FDC across VFC and presents resource allocation and scheduling for the considered real traffic. This is the first work to dynamically schedule incoming traffic into the FNs. Besides, this is the first work to study the impact of a data stream on network resource allocation.
- (ii) A distributed Fog-based VFC architecture is presented that, over moderate-bandwidth, provides high availability operations to mobile users.
- (iii) A case study is defined—“StreamVehicularFog” (SVF)—which is an Internet-assisted peer-to-peer (P2P) service architecture, and the stream of data passed/processed over the underutilized networking/computing servers in FDC is evaluated.
- (iv) The simulation results show a significant reduction in the total energy consumption over FDC with various forms of communication links using real datasets.

The rest of the paper is organized as follows: The background is detailed in Section 2. A description of the FDC architecture is reported in Section 3. A simple case study of the Fog data center data stream management and its correlated problem/solution is reported in Section 4. A performance evaluation of the optimal formulations is reported in Section 5. Finally, Section 6 concludes our work.

2. Related Work

In this section, we aim to focus on the resource management of Cloud/Fog-based distributed computing architectures for the energy-efficient support of real-time Big Data Streaming (BDS) applications by resource-limited wireless devices [13–16]. In detail, the S4 and DSstream management frameworks in [13,14] perform dynamic resource scaling of the available virtualized resources by explicitly accounting for the delay-sensitive nature of the streaming workload offloaded by proximate wireless devices. The time stream and PLAstiCC resource orchestrator in [15,16] integrate dynamic server consolidation and inter-server live VM migration. Overall, as in our contribution, the shared goals of [13–16] are (i) the provision of real-time computing support to BDS applications run by resource-limited wireless devices through the exploitation of the virtualized resources made available by proximate FNs; (ii) minimization of the overall inter/intra-data center computing-plus-networking energy consumption under BDS applications. However, unlike our contribution, (i) the resource management techniques developed in [13,14] are not capable of self-tuning to the (possibly, un-predictable) time fluctuation of the workload to be processed; (ii) the management approaches pursued in [15,16] do not guarantee hard limits on the resulting per-task execution times.

Focusing on FC over the VFC architecture, FC affords proximity to the end users, a dense geographical distribution and supports mobility [3]; this means that it provides services in local areas, near to the users, rather than global services [3,17,18].

The authors of [19] have explored the way in which Fog Computing applications are applied in new fall detection algorithms, as well as implementing them into a distributed fall detection system in inter-FNs. The goal of [20] was the attainment of an optimized delay-vs.- energy trade-off under bag-of-task type vehicular applications that are executed on Dynamic Frequency Voltage Scaling (DVFS)-enabled data centers.

In the vehicular scenarios that are presented in [21,22], FNs are virtualized and connected to the networked data center (VNetFCs); they are hosted by Road Side Unit (RSU) at the edge of the Vehicular Network and the offloading task can be achieved without introducing intolerable delay. In the IoTs era, Foglets (Fls) are defined as large implemented serving demands for the future 5 G [23].

3. FDC over VFC Paradigm

In this paper, a basic system model for VFC with multiple foglets (Fls) is illustrated in Figure 1. The current practice allocates mobile tasks to the closest Fl. The model provides and operates multiple Fls. Moreover, we consider a case study and go in depth in the FDC as a vital component in VFC; we present a general resource allocation model and propose an adaptive scheduler which covers the

hard limited QoS. FC is a platform that uses a collaborative multitude of near-users edge devices [3] with some important characteristics shown in Table 1. FC provides data, computing, storage and application services [3,18] to end users [24]. An example scenario is shown in Figure 1. Specifically, in Figure 1, we adopt three main areas: vehicles area, RSU area, and FDC area [3,17,18].

Table 1. Main characteristics for vehicular Fog computing.

FC	FDC	←FN	←RSU	←Vehicles
Access Medium	Fix Wireless	Fix Wireless	Wireless	Wireless
Technologies		WiFi/3G/4G-LTE		
Mobility	No	No	No	Yes
Proximity	Round	Near	Near	Nearly
Connectivity	Multi Hops	Multi Hops	Single Hop	Single Hop
Geographic Distribution	Low	Medium	High	Very High
Heterogeneity	No	Yes	Yes	Yes
Architecture	Centralized	Distributed	Distributed	Mobile
Bandwidth	High	High	High	Low
Latency	Low	Low	Low	Low
Delay Jitter	Low	Very Low	Very Low	Low
Network Area	Core	Edge	Field	Edge
QoS	Improve	High	High	High

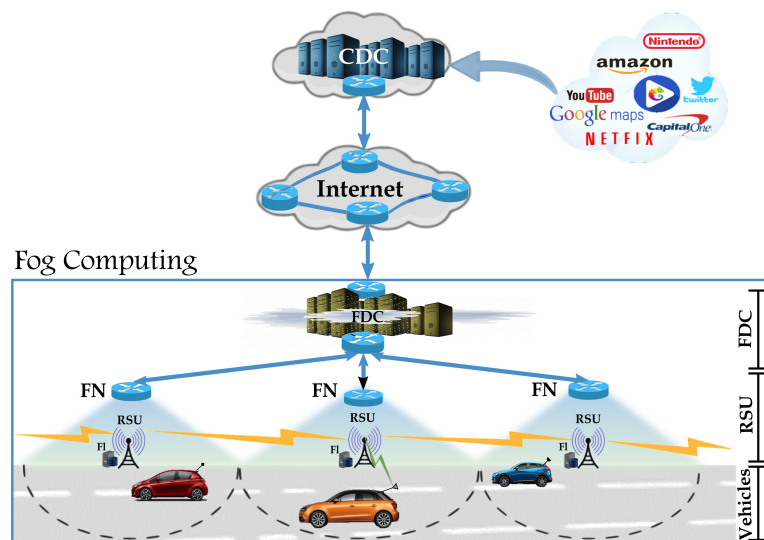


Figure 1. The overall VFC architecture. CDC: Cloud-Data-Center with the main areas; FN: Fog-Node; RSU: Road-Side-Unit; Fl: Foglet; FDC: Fog-Data-Center.

In the architecture presented in Figure 1, VFC provides a solid platform for applications used to provide services to users. VFC services are mainly implemented in Fog: these use a distributed architecture over moderate-bandwidth to provide high availability operations. As we see, vehicles are attached to one of the RSUs in the FN [12], with the heterogeneous type of devices considered as a physical infrastructure. Indeed, each FN consists of a single server or a set of servers that builds heterogeneous requests. On the other hand, vehicles are equipped with a smart device (e.g., smartphones) [7,17] that is wirelessly connected to the RSU. The mobile vehicles arbitrarily enter the field of the RSU and may leave its coverage area. FNs could be interconnected and each of them is linked to the FDC. Vehicles could request a service from the RSU (then transfer this request to the FN and finally the FDC to manage the request) while they freely move across the FDC and RSU areas and the FDC can provide them with the required services in order to facilitate the

connectivity between service providers and users. Application services are provided for near-users that are located in this area (FDC area) as well as direct access to the context parameters with the main VFC characteristics shown in the Table 1. The management architecture of FNs should include challenge strategies for resource allocation, as well as the management of edge devices and computing and storage resources. Also, it is important to note that the architecture should be managed by Software Defined Networks (SDNs) [3] and Network Functions Virtualization (NFV) [24]. The correlated explanations are not within the scope of this paper. Motivated by these considerations, we aim to focus on the Fog computing environment and FDC component which are detailed in the next section.

4. A Case Study: StreamVehicularFog

We now move our attention to a more formal modeling of the presented case study and its correlated models. Motivated by the aforementioned consideration and based on the self-configuring framework as a prototype which has recently been presented in [25], we define a cases study, named StreamVehicularFog (SVF), as a paradigm which is covered in an Internet-assisted peer-to-peer (P2P) service architecture. The SVF aims to minimize the total energy cost in mobile devices and data centers with an adaptive synchronization of the resource management operations that guarantee the stream of data passed/processed over the underutilized networking/computing servers in the FDC. In this scenario, FC is equipped with RSUs to broadcast locally processed data to smartphones in the vehicles through point-to-point TCP/IP -based I2V connections with only one hop wireless link. In order to evaluate the performance of our model, we consider the workload of a real-time stream, as is presented in Figure 2; the arrival rate is a challenging task and should be processed through the FDC to the available servers. Moreover, the FDC adopts some resource allocation and scheduler techniques to analyze the data and set up frequent ON/OFF transitions of the underutilized computing/communication servers with the aim of energy provision. Indeed, in turn, Figure 2 presents the trace of the input/output (I/O) arrival data for the smart-phones in each vehicle of the VFC model in Figure 1. Note that, in this case study, we assume that the processing data are gathered from the vehicular application service requested.

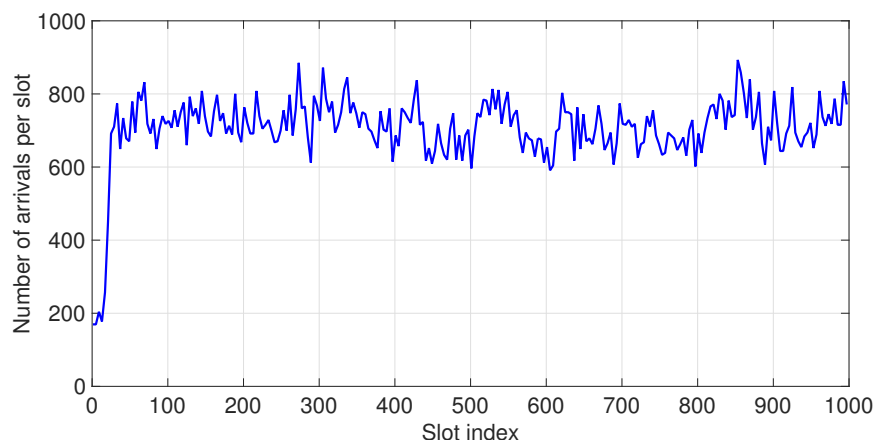


Figure 2. Trace sample of I/O arrival data from an enterprise cluster in WorldCup98 Workload [1:1000] [26]. The corresponding average arrival rate and PMR (Peak Mean Rate) are 1.56 and 19.65, respectively.

In order to understand the FDC structure in the SVF scenario, we modeled it in Figure 3. We understand that each FN sends the incoming traffic to the load dispatcher via the router gateway and transfers the traffic to the available virtual machines (VMs) that run on the servers for the processing. We need to minimize the computation cost in each server as well as decreasing the inter-communication TCP/IP connection cost between VMs. In this model, the virtual machine

manager is responsible for the resource allocation and VM management. Therefore, we need to propose an efficient scheduling method for this resource management problem.

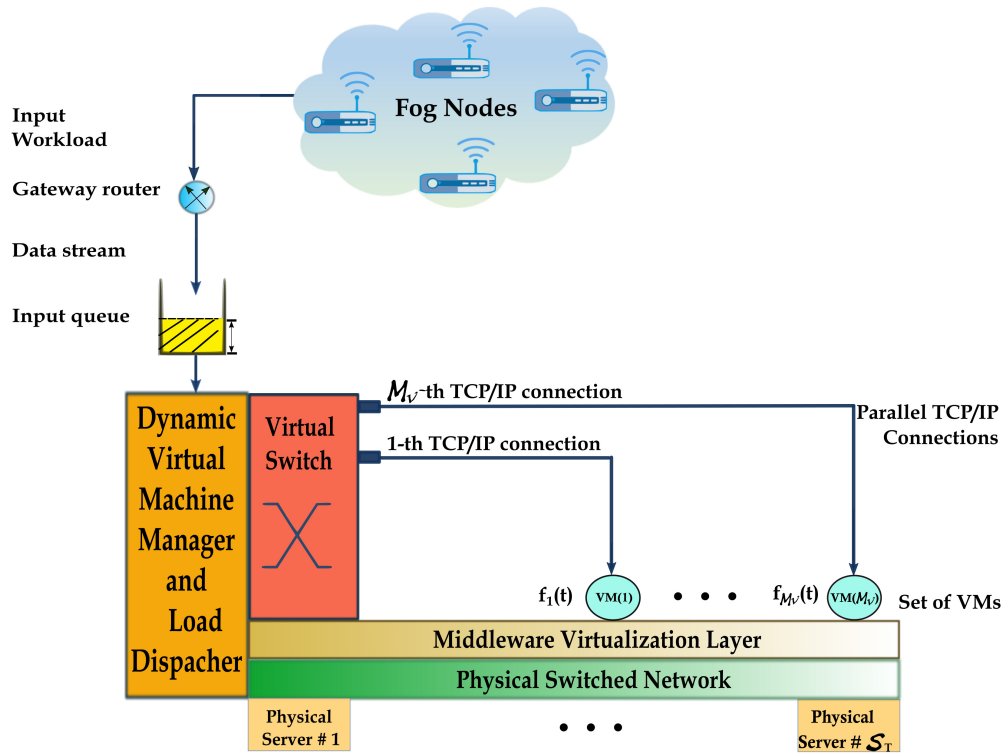


Figure 3. FDC structure in the StreamVehicularFog scenario.

To address this problem, firstly, we formulate a stochastic programming problem considering the FN uncertainties. We propose a general algorithm to solve the simple problem engaged in FDC and give some results. Due to the potentially high number of involved VCs, RSUs, and hosted VMs, we require that the resulting adaptive scheduler allows scalable and distributed implementation in the emerging paradigm of cognitive computing.

4.1. StreamVehicularFog Virtualized TCP/IP Communication Model

As observed in Figure 3, the master functional block of our considered FDC is the intra-communication part. The FDC operates at the Middleware layer and is composed of (i) an Admission Control Server (ACS) that is also considered as a gateway Internet router; (ii) an input buffer of the admitted workload (where our corresponding resource scheduler adaptively performs the control and allocation admission of the available virtual resources) shared by the VMs; (iii) the Load Dispatcher that reconfigures and consolidates the available computing/networking physical resources, dispatching the workload buffered by the input/queue to the turned ON VMs that process the currently assigned task by self-management, thus guaranteeing the demand of the local virtualized computation and communication virtual resources; (iv) the Virtual Switch that controls/manages the TCP/IP (TCPNewReno congestion (by the switches) avoidance state) protocol as an effective means of managing the end-to-end transport connections, allowing the internal VM communication.

4.2. StreamVehicularFog Computing Model

We consider the most general case composed of multiple reconfigurable VMs per physical server interconnected by a switched rate-adaptive virtual LAN (VLAN), managed by the controller, named StreamVehicularFog Manager (SVFM), with the following parameters:

- $\mathcal{S}_T \triangleq$ total number of physical servers with $\mathcal{S}_T \geq 1$;
- $\mathcal{VM}_{max}(s) \triangleq$ maximum number of VMs (possibly, heterogeneous) hosted by the s -th physical server with $\mathcal{VM}_{max}(s) \geq 1$, $s = 1, \dots, \mathcal{S}_T$;
- $\mathcal{VM}(s;v) \triangleq v$ -th VM hosted by the s -th server, $1 \leq v \leq \mathcal{VM}_{max}(s)$, $1 \leq s \leq \mathcal{S}_T$;
- $\mathcal{M}_{sv} \triangleq$ total number of VMs in the data center that is calculated as: $\sum_{s=1}^{\mathcal{S}_T} \mathcal{VM}_{max}(s) \equiv \mathcal{S}_T \times \mathcal{VM}_{max}$. Formally, \mathcal{M}_{sv} is the set of available VMs which may be hosted by the FDC.

Moreover, the engaged internal parameters for the server of the FDC are presented as follows:

- $\mathcal{M}(t) \triangleq$ set of the VMs which are turned ON at (t) time slot $(t) \subseteq \{(s;v); 1 \leq s \leq \mathcal{S}_T; 1 \leq v \leq \mathcal{VM}_{max}(s)\}$
- $\overline{\mathcal{M}}(t) \triangleq$ set of the VMs which are turned OFF at $(t) \subseteq \{(s;v); 1 \leq s \leq \mathcal{S}_T; 1 \leq v \leq \mathcal{VM}_{max}(s)\}$
- $\overline{\mathcal{M}}(t) \equiv \{(s;v); 1 \leq s \leq \mathcal{S}_T; 1 \leq v \leq \mathcal{VM}_{max}(s)\} / \mathcal{M}(t)$;
- $f_{s,v}(t) \triangleq$ processing rate of VM($s;v$) at $t \in [0, f_{s,v}^{max}](IU/s)$, $1 \leq s \leq \mathcal{S}_T$, $1 \leq v \leq \mathcal{VM}_{max}(s)$;
- $\mathcal{L}_{s,v}(t) \triangleq$ workload processed by VM($s;v$) at $t \in [0, \mathcal{L}_{s,v}^{max}](IU)$, $1 \leq s \leq \mathcal{S}_T$, $1 \leq v \leq \mathcal{VM}_{max}(s)$;
- $f_{s,v} \triangleq$ maximum processing rate of VM($s;v$), (IU/s) , $1 \leq s \leq \mathcal{S}_T$, $1 \leq v \leq \mathcal{VM}_{max}(s)$;
- $\mathcal{L}_{s,v}^{max} \triangleq t_{slot} f_{s,v}^{max} \triangleq$ maximum workload processed by VM($s;v$) during each time slot (t) ; in (IU) , $1 \leq s \leq \mathcal{S}_T$, $1 \leq v \leq \mathcal{VM}_{max}(s)$;
 $\mathcal{L}_{s,v}^{max} \leq t_{slot} f_{s,v}^{max}$.

Let $t \in \mathcal{T}$ be a consolidation instant. Each slot t can be considered according to the following states:

1. turn ON some physical servers which were turned OFF at slot t ;
2. turn OFF some physical servers which were turned ON at slot t ;
3. turn ON or turn OFF some VMs hosted by servers which were turned ON at slot $(t-1)$ and are still turned ON at slot t ;
4. scaling up/down the processing rates and workloads of some VMs which are turned ON at slot t .

According to the aforementioned engaged parameters, the following actions should be taken to manage the problem applicable in the real environment: Let $f_{ON}(IU/s)$ be the minimum allowable practical processing rate for turned ON VMs and $\mathcal{M}(t) \equiv \{(s;v) : f_{s,v}(t) > f_{ON}; 1 \leq s \leq \mathcal{S}_T; 1 \leq v \leq \mathcal{VM}_{max}(s)\}$ and $\overline{\mathcal{M}}(t) \equiv \{(s;v) : f_{s,v}(t) \leq f_{ON}; 1 \leq s \leq \mathcal{S}_T; 1 \leq v \leq \mathcal{VM}_{max}(s)\}$.

4.3. Energy Models

Let $\mathcal{E}_c^{idle}(s)[J]$ and $\mathcal{E}_c^{max}(s)[J]$ $1 \leq s \leq \mathcal{S}_T$ be the energies wasted by the s -th physical server in the idle (inactive) state and we have $\mathcal{E}_c^{idle}(s) \leq \mathcal{E}_c^{max}(s)$ and the TCP/IP intra-connection inside the FDC can be considered under Fast/Giga Ethernet with the practical ranges of values for $\Omega_{s,v}^c$ as the bandwidth power coefficient of $[J/(IU)^\gamma]$, and R_{max} as the end-to-end band-width rate in $([IU/sec])$ as follows:

- Fast Ethernet: $1.1 \times 10^{-3} \leq \Omega_{s,v}^c \leq 1.8 \times 10^{-3}$, $R_{max} = 100$
- Giga Ethernet: $1.8 \times 10^{-3} \leq \Omega_{s,v}^c \leq 2.5 \times 10^{-2}$, $R_{max} = 1000$.

The corresponding communication energy $\mathcal{E}_{s,v}^{LAN}[J]$ wasted by the s -virtual link is: $\mathcal{E}_{s,v}^{LAN} \equiv \mathcal{E}_{s,v}^{LAN}(\mathcal{L}_{s,v}) = \Omega_{s,v}^c(\mathcal{L}_{s,v})^\gamma[J]$, $1 \leq v \leq \mathcal{VM}_{max}(s)$; $s = 1, \dots, \mathcal{S}_T$.

When the VMs hosted per physical server are perfectly isolated, the resulting model for the overall wasted computing energy simplifies to the following formula:

$$\mathcal{E}_T(t) = \mathcal{E}_C(t) + \mathcal{E}_S(t) + \mathcal{E}_N(t) [J]$$

where $\mathcal{E}_C(t)$ is the SVF computing energy; $\mathcal{E}_S(t)$ is the SVF energy switching; and, $\mathcal{E}_N(t)$ is the SVF TCP/IP communication energy. Therefore, the \mathcal{E}_T for slot t is defined as

$$\mathcal{E}_T(t) = \sum_{s=1}^{S_T} \left\{ \left[\mathcal{E}_c^{idle}(s) \left(\sum_{v=1}^{\mathcal{VM}_{max}(s)} f_{s,v}(t) \right) + \left(\frac{\mathcal{E}_{c,v}^{max}(s) - \mathcal{E}_{c,v}^{idle}(s)}{\mathcal{VM}_{max}(s)} \right) \left(\sum_{v=1}^{\mathcal{VM}_{max}(s)} \left(\frac{f_{s,v}(t)}{f_{s,v}^{max}} \right)^w \right) \right] + \right. \\ \left. k_e \left(\sum_{v=1}^{\mathcal{VM}_{max}(s)} (f_{s,v}(t) - f_{s,v}(t-1))^2 \right) + \left(\sum_{v=1}^{\mathcal{VM}_{max}(s)} \Omega_{s,v}^c (\mathcal{L}_{s,v})^\gamma(t) \right) \right\}, \quad (1)$$

where $t \in T, t \geq 1$; γ is the powered model for the energy held in the per-transport layer connections per energy instant t ; Ω — is the common energy consumptions of all LAN (TCP/IP) connections measured $[J/(IU)^\gamma]$; and $w \geq 2$ is strictly convex in $f_{s,v}(t)$. Therefore, the general optimization problem can be simplified as

$$\min\{\mathcal{E}_T(t)\}, \quad (2)$$

subjected to:

$$\mathcal{L}_T(t) - \left(\sum_{s=1}^{S_T} \sum_{v=1}^{\mathcal{VM}_{max}(s)} \mathcal{L}_{s,v}(t) \right) = 0, \quad (3)$$

$$\mathcal{L}_{s,v}(t) - t_{slot} f_{s,v}^{max} \leq 0, \quad \forall s, v \in \mathcal{VM}(s; v), \quad (4)$$

$$f_{s,v}(t) - f_{s,v}^{max} \leq 0, \quad \forall s, v \in \mathcal{VM}(s; v), \quad (5)$$

where t_{slot} is the hard deadline considered to process each slot; in fact, it can be considered as the service level agreement for the mobile application requested from VFC. The workload processed should not be more than the input workload in each time slot (t) (see Equation (3)), and the frequency of each VM should not be more than the maximum frequency (see Equations (4) and (5)).

In order to resolve the problem, we can consider the problem as a *bin packing penalty-aware* problem. Algorithm 1 presents the pseudocode of the proposed solution. Indeed, in turn, for each incoming workload $\mathcal{L}_T(t)$, we calculate how many VMs can be allocated for the t -th slot. After that, we consider servers as bins and VMs as packs that must be distributed based on their time limitations and frequency limitations. For each iteration it , we give a penalty to each server based on its energy characteristics (idle energy, maximum energy, maximum frequency) and look for the best servers to be allocated. The penalty is considered in order to decrease the number of fatality cases (increasing the energy consumptions exponentially). This means that the server will be punished and banned from being used again in the next it . When the freezing iterations n -th for the server have passed, the server will return to the list of available servers for the next it . Note that the server S can serve VMs when it does not pass the maximum number of VMs $\mathcal{VM}_{max}(s)$. Otherwise, the server S pulls out from the list of processing servers. The maximum interactions are defined $\mathcal{M}_{s,v}$. The process is carried out for each time slot t until all the incoming workloads have been served. Finally, we calculate the energy consumption components of the optimization problem for each slot t .

Algorithm 1 Pseudo-code Heuristic Solution.**INPUT:** $\mathcal{S}_T, \hat{\mathcal{S}}, \mathcal{M}_{sv}, \mathcal{L}_T, f_{s,v}^{max}, \mathcal{E}_{c,v}^{idle}, \mathcal{E}_{max}, \mathcal{T}$ **OUTPUT:** $\mathcal{E}_T, \hat{\mathcal{S}}(t), \mathcal{M}(t)$

```

1: for  $t \geq 1$  do            $t \in T$ 
2:   Check feasibility of input
3:   for  $it = 1 : \mathcal{M}_{sv}$  do
4:     if  $\mathcal{L}(t) \leq f_{s,v}^{max} t_{slot}$  then
5:        $f_{s,v}(t) = \mathcal{L}(t) / t_{slot}$ ;
6:                                     ▷ Update set of servers  $\hat{\mathcal{S}}(t)$  with 0 penalty;
7:       Find the minimum  $\mathcal{E}_{c,v}^{idle}(t)$  in  $\hat{\mathcal{S}}(t)$ ;
8:       if ( $flag(t) == 1$ ) then
9:                                     ▷ decrease the penalty (-1) in all the servers with  $penalty \geq 1$ ;
10:      end if
11:                                     ▷ add new VM into the  $\hat{\mathcal{S}}(t)$ ;
12:    end if
13:  end for
14:  Update  $\hat{\mathcal{S}}(t)$  and  $\mathcal{M}(t)$  through  $\mathcal{VM}(s;v)$ ;
15:  Calculate  $\mathcal{E}(t)$  as in (1);
16: end for
17: return  $\mathcal{E}_T, \hat{\mathcal{S}}(t), \mathcal{M}(t)$ 

```

5. Performance Evaluation

In this section, we aim to test the proposed method on the FDC.

5.1. Test Scenario

In order to guarantee the accuracy and timeliness of the VC information transmission, it is necessary to analyze and act upon the data in less than a second. Data are sent to the Cloud for historical analysis, BD analytics, and long-term storage with less latency. For example, each of the thousands or hundreds of thousands of FNs might send periodic summaries of the data grid to the Cloud for historical analysis and storage.

In the test carried out, the FNs are assumed to be placed over a spatially limited Intranet. We consider a physical topology with four heterogeneous FNs. In order to model the CPU consumptions of the FNs, we refer to the Intel R-CoreTM 2 CPU Q6700 with 2.67 GHz frequency rate and 4 GB of RAM memory. In order to confirm the correctness of the proposed infrastructure in Figure 1, we require an evaluation to explore several resource management and scheduling techniques such as operator, application, task placement and consolidation. We evaluate some resource management policies applicable to the Fog environment with respect to their impact on latency, network usage, and energy consumption. In this framework, the VCs publish data to SVF; applications are run on Fog devices (FN) to process data coming from VCs. Indeed, it is important to test the solution with a real-world input workload and compare it with the corresponding solutions of the recent FDC-based techniques. So, in the next subsection, we will move on to this issue and address it with several simulated testbed cases.

5.2. Test Result

In this subsection, we aim to test our scheduler on 100 incoming data streams ($T = 100$) for a Fast Ethernet communication channel and calculate the on-line instantaneous overall energy consumption and energy savings that are shown in Figure 4. From this figure, we conclude that the total energy cost changes over time; the figure even shows the energy saving correlated to time.

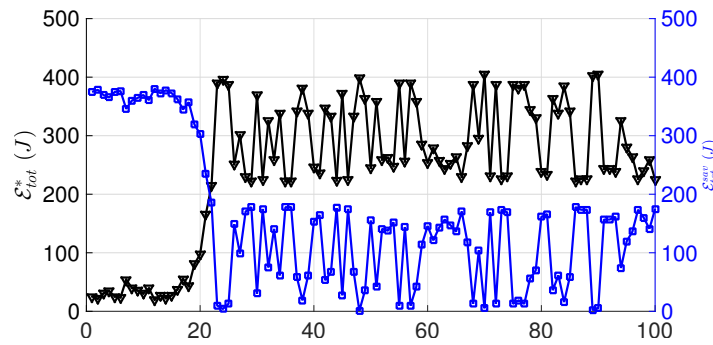


Figure 4. Energy saving of SVF (dashed plot) and energy consumption of SVF (continuous plot) for the Fast Ethernet LAN for $T = 100$.

On the other hand, in order to evaluate the instantaneous energy consumption for 1000 slots using two different schemes of the communication network (Fast and Giga), we evaluated the Fast/Giga Ethernet energy consumption in the SVF scenario. The results of these comparisons are presented in Figure 5. We understand that when the $\Omega_{s,v}^c$ increases, the correlated energy consumption rises at the same scale due to the rising communication constant; this trend continues over time.

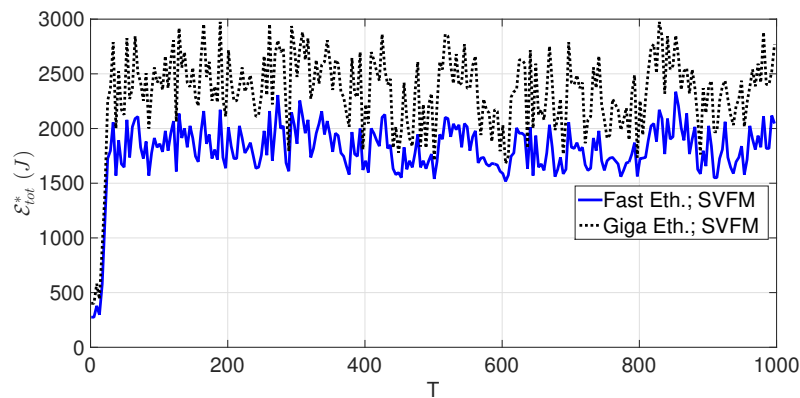
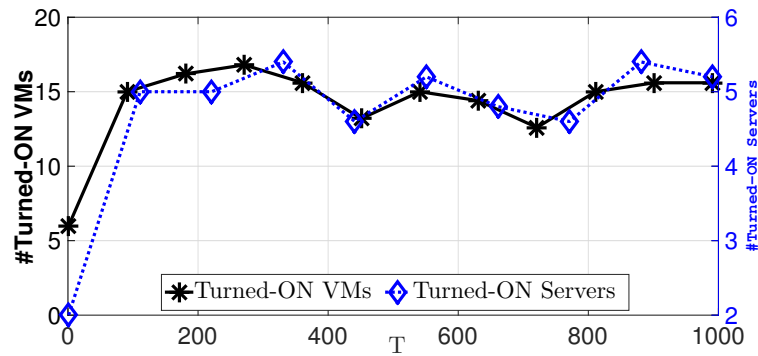
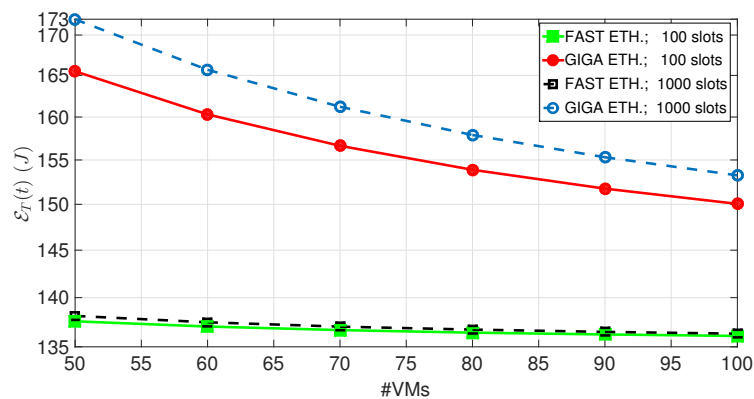


Figure 5. Fast/Giga Ethernet LANs on the energy consumption of SVF for $T = 1000$.

Finally, in the last test scenario, the plots of Figure 6a show the time behaviors in relation to the number of VMs and physical servers that are dynamically turned ON by our algorithm scheduler under the real-world workload of Figure 2. A comparison of the plots of Figure 6a with the arrival trace of Figure 2 supports the conclusion that the consolidation action performed by the proposed algorithm is capable of tracking the abrupt (and unpredicted) changes of the input workload with a delay of 1 slot period. An examination of Figure 6a reveals that (i) the number of turned ON VMs/servers increases with the increasing values of timeslots (T) (i.e., the network state pushes the scheduler to keep more FNs and corresponding servers turned ON and responds to the injected traffic over the FIs); and, (ii) the rate of turned ON servers in each timeslot is higher than the rate of turned ON VMs; this confirms that the servers are consuming more energy than VMs and when we use VMs we can control the server energy consumption. Also, this rate depends on the adopted Ethernet technology applied. Focusing on Figure 2, it is interesting to note that the numerically evaluated average energy consumption over 100 VMs (10 servers residing in four heterogeneous FNs) of the proposed heuristic for 100 slots is about 136.5 (J) and 138.5 (J); for 1000 slots, this value is about 155 (J) and 160 (J) for GIGA and FAST Ethernet communication cases, respectively. This confirms that even increasing the slots to 10 times the average energy consumption in the proposed scheduler will not lead to a significant increase—at most, 23% in the worst case scenario.



(a) VMs turned ON-vs.- Timeslot (incoming traffic).



(b) Total energy consumption on the FDC-vs.- number of running VMs per FN.

Figure 6. (a) Behavior time in relation to the number of turned ON VMs and physical servers, and (b) the average total energy in $T = 1000$, respectively.

6. Conclusions and Future Developments

The ability of Fog Computing is limitless. It has the potential to solve many problems in various, as yet unexplored fields such as context-aware data production/transfer in vehicular networks. Fog computing over vehicular networks is a platform that offers real-time services to clients. In this paper, we have studied one of the most significant issues of FC, i.e., Fog data center data streams requested by vehicular clients. We proposed a holistic VFC architecture, focused on the FDC data streams; we formulated the problem analytically and resolved the presented resource allocation and scheduling algorithm over a simplified case study: StreamVehicularFog (SVF). In this regard, the SVF simulations, both at small and large scales, confirmed the considerable reduction in the average energy cost that can be achieved by scheduling in mobile Edge/Fog computing of SVF. We tested the presented case study in various test scenarios and confirmed the energy efficiencies. Due to the energy-efficient adaptive management of the delay-vs.-throughput, the trade-off of the SVF TCP/IP mobile connections is an additional topic for further research.

Acknowledgments: This work has been partially supported by the PRIN2015 project with grant number 2015YPMH4W_004: “A green adaptive Fog computing (FC) and networking architecture (GAUCHO)”, funded by the Italian MIUR and by the project “Vehicular Fog energy efficient quality of service (QoS) mining and dissemination of multimedia Big Data streams (V-Fog)” funded by Sapienza University of Rome. Moreover, Mauro Conti is supported by a Marie Curie Fellowship funded by the European Commission (agreement PCIG11-GA-2012-321980). This work is also partially supported by the EU TagItSmart! Project (agreement H2020-ICT30-2015-688061), the EU-India REACH Project (agreement ICI+/2014/342-896), and by the projects “Physical-Layer Security for Wireless Communication”, and “Content Centric Networking: Security and Privacy Issues” funded by the University of Padua. This work is partially supported by the grant No. 2017-166478 (3696) from Cisco University Research Program Fund and Silicon Valley Community Foundation.

Author Contributions: All authors together proposed, discussed and finalized the main idea of this work. Paola G. V. Naranjo and Zahra Pooranian proposed the idea, calculated and plotted the feasible regions, and implemented the algorithm and their comparisons. Shahaboddin Shamshirband, Jemal H. Abawajy and Mauro Conti helped in the paper preparations, English corrections and submission.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Karagiannis, G.; Altintas, O.; Ekici, E.; Heijenk, G.; Jarupan, B.; Lin, K.; Weil, T. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 584–616.
2. Naranjo, P.G.V.; Pooranian, Z.; Shojafar, M.; Conti, M.; Buyya, R. FOCAN: A Fog-supported Smart City Network Architecture for Management of Applications in the Internet of Everything Environments. *arXiv* **2017**, arXiv:1710.01801.
3. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the internet of things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 17 August 2012; pp. 13–16.
4. Mahmud, R.; Kotagiri, R.; Buyya, R. Fog computing: A taxonomy, survey and future directions. In *Internet of Everything*; Springer: Berlin, Germany, 2018; pp. 103–130.
5. Tajiki, M.M.; Salsano, S.; Shojafar, M.; Chiaraviglio, L.; Akbari, B. Joint Energy Efficient and QoS-aware Path Allocation and VNF Placement for Service Function Chaining. *arXiv* **2017**, arXiv:1710.02611.
6. Shojafar, M.; Cordeschi, N.; Baccarelli, E. Energy-efficient adaptive resource management for real-time vehicular cloud services. *IEEE Trans. Cloud Comput.* **2016**, 1–14, doi:10.1109/TCC.2016.2551747.
7. Büsching, F.; Schildt, S.; Wolf, L. DroidCluster: Towards Smartphone Cluster Computing—The Streets are Paved with Potential Computer Clusters. In Proceedings of the 2012 32nd International Conference on Distributed Computing Systems Workshops, Macau, China, 18–21 June 2012; pp. 114–117.
8. D’Andreagiovanni, F.; Mett, F.; Nardin, A.; Pulaj, J. Integrating LP-guided variable fixing with MIP heuristics in the robust design of hybrid wired-wireless FTTx access networks. *Appl. Soft Comput.* **2017**, *61*, 1074–1087.
9. Wirtz, T.; Ge, R. Improving mapreduce energy efficiency for computation intensive workloads. In Proceedings of the 2011 International Green Computing Conference and Workshops (IGCC), Orlando, FL, USA, 25–28 July 2011; pp. 1–8.
10. Pooranian, Z.; Shojafar, M.; Naranjo, P.G.V.; Chiaraviglio, L.; Conti, M. A Novel Distributed Fog-Based Networked Architecture to Preserve Energy in Fog Data Centers. In Proceedings of the 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Orlando, FL, USA, 22–25 October 2017; pp. 604–609.
11. Dabbagh, M.; Hamdaoui, B.; Guizani, M.; Rayes, A. An energy-efficient VM prediction and migration framework for overcommitted clouds. *IEEE Trans. Cloud Comput.* **2016**, doi:10.1109/TCC.2016.2564403.
12. Aazam, M.; Huh, E.N. Fog computing and smart gateway based communication for cloud of things. In Proceedings of the 2014 International Conference on Future Internet of Things and Cloud (FiCloud), Barcelona, Spain, 27–29 August 2014; pp. 464–470.
13. Neumeyer, L.; Robbins, B.; Nair, A.; Kesari, A. S4: Distributed stream computing platform. In Proceedings of the 2010 IEEE International Conference on Data Mining Workshops (ICDMW), Sydney, Australia, 13 December 2010; pp. 170–177.
14. Zaharia, M.; Das, T.; Li, H.; Shenker, S.; Stoica, I. Discretized Streams: An Efficient and Fault-Tolerant Model for Stream Processing on Large Clusters. *HotCloud* **2012**, *12*, 10.
15. Qian, Z.; He, Y.; Su, C.; Wu, Z.; Zhu, H.; Zhang, T.; Zhou, L.; Yu, Y.; Zhang, Z. Timestream: Reliable stream computation in the cloud. In Proceedings of the 8th ACM European Conference on Computer Systems, Prague, Czech Republic, 15–17 April 2013; pp. 1–14.
16. Kumbhare, A.G.; Simmhan, Y.; Prasanna, V.K. Plasticc: Predictive look-ahead scheduling for continuous dataflows on clouds. In Proceedings of the 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), Chicago, IL, USA, 26–29 May 2014; pp. 344–353.

17. Tang, B.; Chen, Z.; Hefferman, G.; Wei, T.; He, H.; Yang, Q. A hierarchical distributed fog computing architecture for big data analysis in smart cities. In Proceedings of the ASE BigData & SocialInformatics, Kaohsiung, Taiwan, 7–9 October 2015; p. 28.
18. Skala, K.; Davidovic, D.; Afgan, E.; Sovic, I.; Sojat, Z. Scalable distributed computing hierarchy: Cloud, fog and dew computing. *Open J. Cloud Comput.* **2015**, *2*, 16–24.
19. Cao, Y.; Chen, S.; Hou, P.; Brown, D. FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation. In Proceedings of the 2015 IEEE International Conference on Networking, Architecture and Storage (NAS), Boston, MA, USA, 6–7 August 2015.
20. Kim, K.H.; Buyya, R.; Kim, J. Power Aware Scheduling of Bag-of-Tasks Applications with Deadline Constraints on DVS-enabled Clusters. *CCGrid* **2007**, *7*, 541–548.
21. Bitam, S.; Mellouk, A.; Zeadally, S. VANET-cloud: A generic cloud computing model for vehicular Ad Hoc networks. *IEEE Wirel. Commun.* **2015**, *22*, 96–102.
22. Mtibaa, A.; Fahim, A.; Harras, K.A.; Ammar, M.H. Towards resource sharing in mobile device clouds: Power balancing across mobile devices. *ACM SIGCOMM Comput. Commun. Rev.* **2013**, *43*, 51–56.
23. Chen, M.; Zhang, Y.; Li, Y.; Mao, S.; Leung, V.C. EMC: Emotion-aware mobile cloud computing in 5G. *IEEE Netw.* **2015**, *29*, 32–38.
24. Vaquero, L.M.; Roderio-Merino, L. Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 27–32.
25. Barbera, M.V.; Kosta, S.; Mei, A.; Perta, V.C.; Stefa, J. Mobile offloading in the wild: Findings and lessons learned through a real-life experiment with a new cloud-aware system. In Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 2355–2363.
26. Urgaonkar, B.; Pacifici, G.; Shenoy, P.; Spreitzer, M.; Tantawi, A. Analytic modeling of multitier internet applications. *ACM Trans. Web* **2007**, *1*, doi:10.1145/1232722.1232724.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).