

## Article

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

Paola G. Vinueza Naranjo <sup>1</sup>, Zahra Pooranian <sup>2</sup> , Shahaboddin Shamshirband <sup>3,4,\*</sup>  and Jemal Abawajy <sup>5</sup> 

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

<sup>2</sup> Department of Mathematics, University of Padova, Italy; zahra@math.unipd.it

<sup>3</sup> Department of Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam

<sup>4</sup> Department of Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam; shahaboddin.shamshirband@tdt.edu.vn

<sup>5</sup> Faculty of Science, Engineering and Built Environment, Deakin University, Australia; jemal.abawajy@deakin.edu.au

\* Correspondence: zahra@math.unipd.it; shahaboddin.shamshirband@tdt.edu.vn; Tel.: +39-327-114-4158

**Abstract:** In this paper, we discuss the most significant application opportunities and outline the challenges in performing a real-time and energy-efficient management of the distributed resources available at mobile devices and 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:** big data streaming; energy-efficiency; resource management; Infrastructure-to-Vehicular (I2V); Vehicular Fog Computing (VFC)

## 1. Introduction and Related Work

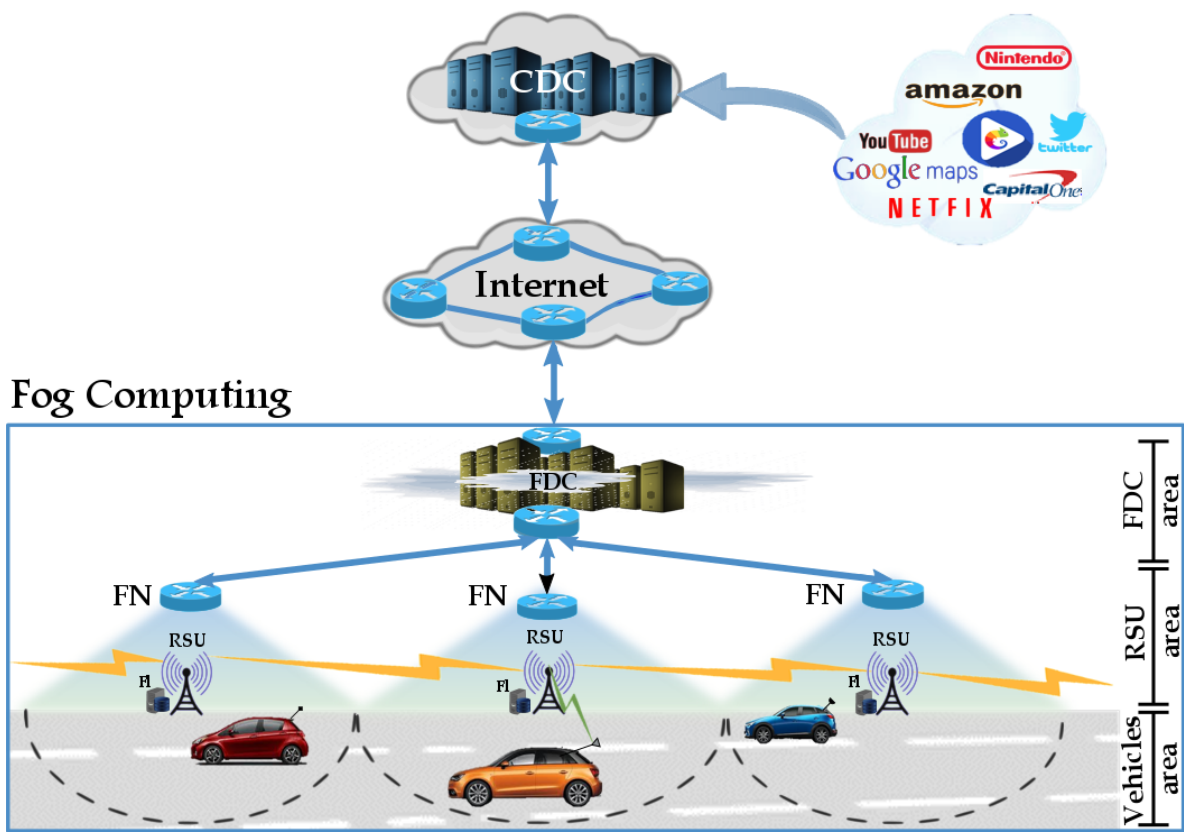
Vehicular Fog Computing (VFC) is a promising paradigm (still unexplored), where the aim is supporting services in order to include the come up Internet infotainment applications. The Cloud Computing (CC) has attract the big companies (such as VTube, Netflix, Cameleon, etc) [1,2] that exploit the Fog Nodes (FNs) of the Infrastructure-to-Vehicle (I2V) with a single-hop IEEE 802.11 likes wireless links for data dissemination, proving low energy consumption [2]. Furthermore, Fog Computing (FC) is used to design an efficient solution within supporting low-latency [3] and energy-efficient as well as self-adaption. It plays as spectrum efficient by means of a strict collaboration among end-devices in the same integrate platform. FC aims at distributing self-powered data center (e.g., FNs) between resource remote clouds and resource-limited smartphone equipped in the vehicles, in order to perform context-aware energy-efficient data mining and dissemination for the adoption of smartphone device and clients' heavy dependence mobile applications [3,4]. Moreover, being deployed in proximity of the Vehicular Clients (VCs), the FNs efficiently exploit the awareness of the connection states supporting latency and delay-jitter. In Vehicular scenarios that is presented in [5,6], FNs are virtualized and connected to networked data center (VNetFCs), which are hosted by Road Side Unit (RSU) at the edge of Vehicular Network and the task offloading can be achieved without introducing intolerable delay. In the IoTs era, Foglets (Fls) are defined as large implemented serving demands for the future 5G [7]. In

29 this paper, a basic system model for VFC with multiple Fls is illustrated in Fig. 1. The current practice  
30 allocates mobile tasks to the closest Fl. The model provides and operates multiples Fls. Moreover,  
31 we consider a case study and go in depth in the Fog Data Center (FDC) as a vital component in VFC,  
32 present general resource allocation model and propose an adaptive scheduler which covers the hard  
33 limited QoSs.

34 The rest of the paper is organized as follow. The description of the FDC architecture is reported in  
35 the next section. A simple case study of the Fog data center data stream managements and its correlated  
36 problem/solution are reported in Sec. 3. The performance evaluation of the optimal formulations is  
37 reported in Sec. 4. Finally, Sec. 5 concludes our work.

38 **2. Fog Computing Paradigm**

39 Fog Computing (FC) is a platform that uses a collaborative multitude of near-users edge devices  
40 [3] with some important characteristics shown in Table 1. It provides data, computing, storage and  
41 application services [3,8] to end users [9]. FC affords proximity to the end users, a dense geographical  
42 distribution and supports the mobility [3], it means, it provides services in local area instead of global  
43 and near to the users [3,8,10]. An example scenario is shown in Fig. 1. Specifically, in the Fig. 1, we  
adopt a simple three main areas: Vehicles area, RSU area, FDC area [3,8,10].



**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.

44 In this architecture, the VFC provides a solid platform for applications used to provide services  
45 to users. VFC services are mainly implemented in the Fog: these use a distributed architecture over  
46 moderate-bandwidth to provide high availability operations. As we see, vehicles are attached to one  
47 of the RSU in the FN [11], with heterogeneous type of devices considered as a physical infrastructure.  
48 Indeed, each FN consists of a single server or a set of servers that builds heterogeneous requests. On  
49 the other hand, vehicles equipped with a smart device (e.g., smartphones) [4,10] that are wirelessly  
50

connected to the RSU. The vehicles in mobility are arbitrarily entering in the field of RSU and may leave the coverage area of it. FNs could be interconnected and each of them is linked to the FDC. Vehicles could request a service to RSU (i.e, then transfer this request to FN and finally FDC for managing the request) while they are freely moving across the FDC and RSU areas and FDC can provide required services to them in order to facilitate the connectivity between service providers and users. They provide application services for near-users that are located in this area (FDC area) and directly access to the context parameters with the main characteristics of VFC showing in the Table 1. The management architecture of FNs should include challenge strategies for resource allocation, edge devices management jointly with computing and storage resources. Also, it is important to remind that, it should be managed by Software Defined Networks (SDNs) [3] and Network Functions Virtualization (NFV) [9]. The correlated explanations are beyond of the topic 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.

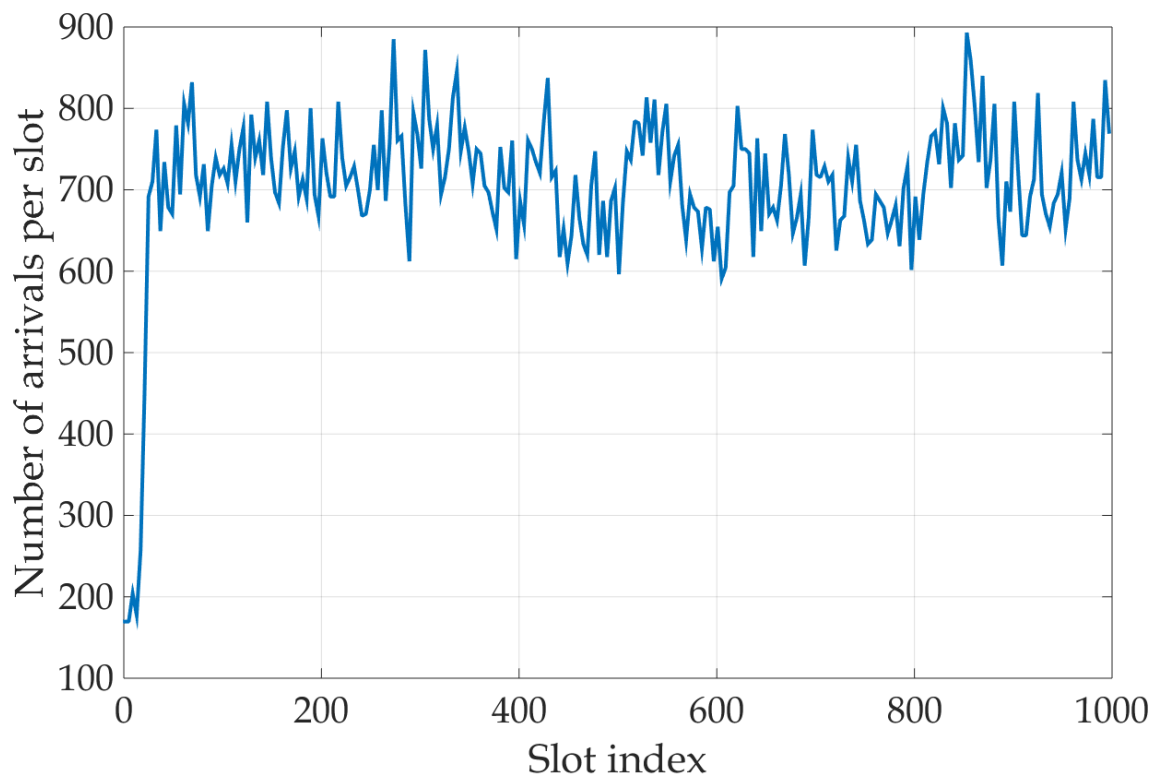
**Table 1.** Main characteristics of Vehicular Fog Computing (VFC).

FC	FDC	FN	RSU	Vehicles
Access Medium	Fix Wireless	Fix Wireless	Wireless	Wireless
Technologies	WiFi/3G/4G-LTE/5G			
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

### 3. A Case Study: StreamVehicularFog

We then move our attention to a more formal modeling of the presented case study and their correlated models. Motivated by the aforementioned consideration and based on the self-configuring framework as a prototype which is recently presented in [12], 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 aim is to minimize the total energy cost in mobile devices and data centers with an adaptive synchronization of the resource management operations that guarantee stream of data passed/process over the underutilized networking/computing servers in FDC. In this scenario, FC 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 a real-time streams workload as is presented in Fig. 2 being the arrival rate for the challenge task that sent to the FNs and should be processed through FDC to the available servers. Moreover, FDC adopts some resource allocation and scheduler techniques to analyze the data and setup frequently ON/OFF transitions of the underutilized computing/communication servers with the aim of energy provisioning. Indeed, in turn, Fig. 2 presents the trace of the I/O arrival data for the smart-phones on each vehicle of the VFC model in

80 Fig. 1. Remind that, in this case study we assume the processing data are gathered from the vehicular application service requested.



**Figure 2.** Trace sample of I/O arrival from enterprise cluster in WorldCup98 Workload [1:1000] [13]. The corresponding average arrival rate and PMR (Peak Mean Rate) are 1.56 and 19.65.

81 In order to understand the FDC structure over SVF scenario, we modeled it in Fig. 3. We  
 82 understand that, each FN sends the coming traffic through the router gateway to the load dispatcher  
 83 and transfer the traffic to the available virtual machines (VMs) running on the servers for the processing.  
 84 We need to minimize the computation cost in each server as well as decreasing the inter-communication  
 85 TCP/IP connection cost between VMs. IN this model, virtual machine manager is responsible for  
 86 the resource allocation and VM management. Therefore, we need to propose an efficient scheduling  
 87 method for this resource management problem. To cope with this problem, firstly, we formulate a  
 88 stochastic programming problem with the considerations of uncertainties in FNs. We propose a general  
 89 algorithm to solve the simple problem engaged in FDC and give some results. Due to the possible  
 90 high number of involved VCs, RSU, and hosted VMs, we require that the resulting adaptive scheduler  
 91 allows scalable and distributed implementation in the emerging paradigm of the cognitive computing.

### 93 3.1. StreamVehicularFog virtualized TCP/IP communication model

94 Reporting Fig.3, the master functional block of our considered FDC is the intra-communication  
 95 part. FDC operates at Middleware layer and is composed by: i) an Admission Control Server (ACS)  
 96 that is considered also as gateway Internet router, ii) an input buffer of the admitted workload (where  
 97 our corresponding resource scheduler adaptively performs the control and allocation admission of the  
 98 available virtual resources) shared by the VMs, iii) the Load Dispatcher that reconfigure and consolidate  
 99 the available computing/networking physical resources, dispatching the workload buffered by the  
 100 input/queue to the turned ON VMs that processes the currently assigned task by self-managing  
 101 guarantee the demand of the local virtualized computation and communication virtual resources,  
 102 iv) the Virtual Switch that control/manage the TCP/IP (TCPNewReno congestion (by the switches)

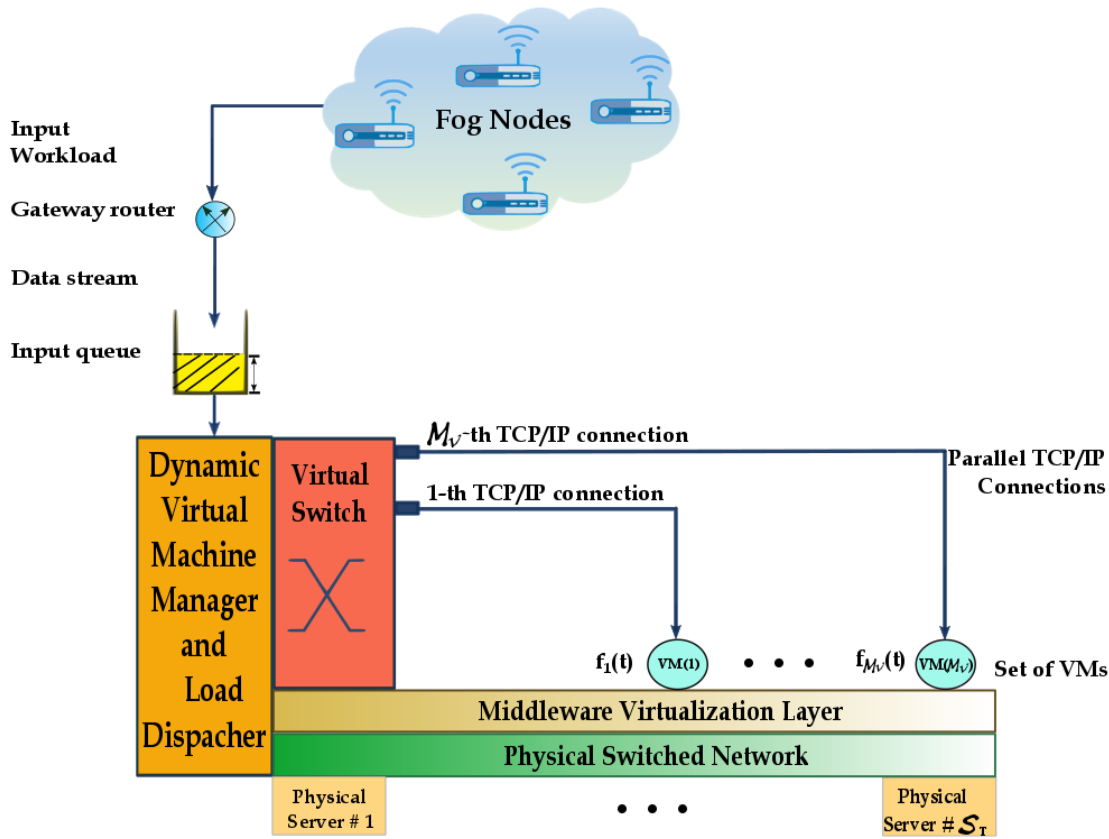


Figure 3. FDC Structure over StreamVehicularFog scenario.

avoidance state) protocol as an effective means to manage the end-to-end transport connections, allowing the internal VM communication.

### 3.2. StreamVehicularFog computing model

We consider the most general case composed by 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:

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

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

- $\mathcal{M}(t) \triangleq$  set of the VMs which are turned-ON at  $(t)$  time slot  $(t) \subseteq \{(s; v); 1 \leq s \leq S_T; 1 \leq v \leq \mathcal{VM}_{max}(s)\}$
- $\overline{\mathcal{M}}(t) \triangleq$  set of the VMs which are turned-OFF at  $(t) \subseteq \{(s; v); 1 \leq s \leq S_T; 1 \leq v \leq \mathcal{VM}_{max}(s)\}$

so that's it

- 122 •  $\overline{\mathcal{M}}(t) \equiv \{(s;v); 1 \leq s \leq \mathcal{S}_T; 1 \leq v \leq \mathcal{VM}_{max}(s)\} / \mathcal{M}(t)$ ;
- 123 •  $f_{s,v}(t) \triangleq$  processing rate of VM(s;v) at  $t \in [0, f_{s,v}^{max}](IU/s)$ ,  $1 \leq s \leq \mathcal{S}_T$ ,  $1 \leq v \leq \mathcal{VM}_{max}(s)$ ;
- 124 •  $\mathcal{L}_{s,v}(t) \triangleq$  workload processed by VM(s;v) at  $t \in [0, \mathcal{L}_{s,v}^{max}](IU)$ ,  $1 \leq s \leq \mathcal{S}_T$ ,  $1 \leq v \leq \mathcal{VM}_{max}(s)$ ;
- 125 •  $f_{s,v} \triangleq$  maximum processing rate of VM(s;v),  $(IU/s)$ ,  $1 \leq s \leq \mathcal{S}_T$ ,  $1 \leq v \leq \mathcal{VM}_{max}(s)$ ;
- 126 •  $\mathcal{L}_{s,v}^{max} \triangleq t_{slot} f_{s,v}^{max} \triangleq$  maximum workload processed by VM(s;v) during each time slot (t); in
- 127  $(IU)$ ,  $1 \leq s \leq \mathcal{S}_T$ ,  $1 \leq v \leq \mathcal{VM}_{max}(s)$ ;
- 128  $\mathcal{L}_{s,v}^{max} \leq t_{slot} f_{s,v}^{max}$ .

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

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

136 According the aforementioned engaged parameters, the following actions should be made to

137 manage the problem applicable in the real environment: Let:  $f_{ON}(IU/s)$  be the minimum practical

138 processing rate allowed a turned-ON VMs and  $\mathcal{M}(t) \equiv \{(s;v) : f_{s,v}(t) > f_{ON}; 1 \leq s \leq \mathcal{S}_T; 1 \leq v \leq$

139  $\mathcal{VM}_{max}(s)\}$  and  $\overline{\mathcal{M}}(t) \equiv \{(s;v) : f_{s,v}(t) \leq f_{ON}; 1 \leq s \leq \mathcal{S}_T; 1 \leq v \leq \mathcal{VM}_{max}(s)\}$ .

### 140 3.3. Energy models

141 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

142 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

143 the FDC can be considered under Fast/Giga Ethernet with the practical ranges of values for  $\Omega_{s,v}^c$  as

144 bandwidth power coefficient in  $[Joule/(IU)^\gamma]$ , and  $R_{max}$  as end-to-end band-witch rate in  $([IU/sec])$

145 as follows:

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

148 The corresponding communication energy  $\mathcal{E}_{s,v}^{LAN}[J]$  wasted by the  $s$ -th virtual link is:  $\mathcal{E}_{s,v}^{LAN} \equiv$

149  $\mathcal{E}_{s,v}^{LAN}(\mathcal{L}_{s,v}) = \Omega_{s,v}^c (\mathcal{L}_{s,v})^\gamma [J]$ ,  $1 \leq v \leq \mathcal{VM}_{max}(s)$ ;  $s = 1, \dots, \mathcal{S}_T$ .

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

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



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

$$\min\{\mathcal{E}_T(t)\}, \quad (2.1)$$

subjected to:

$$\mathcal{L}_T(t) - \left( \sum_{s=1}^{S_T} \sum_{v=1}^{\mathcal{VM}_{max}(s)} \mathcal{L}_{s,v}(t) \right) = 0, \quad (2.2)$$

$$\mathcal{L}_{s,v}(t) - t_{slot} f_{s,v}^{max} \leq 0, \quad \forall s, v \in \mathcal{VM}(s; v), \quad (2.3)$$

$$f_{s,v}(t) - f_{s,v}^{max} \leq 0, \quad \forall s, v \in \mathcal{VM}(s; v). \quad (2.4)$$

150 where  $t_{slot}$  is the hard deadline considered to process each slot, in fact, it can be considered as the  
 151 service lever agreement for the mobile application requested from VFC. The workload processed  
 152 should not be more than input workload in each time slot ( $t$ ) (see eq.(2.2)), and the frequency of each  
 153 VMs should not be more than the maximum frequency (see eqs.(2.3) and (2.4)).

---

**Algorithm 1** Pseudo-code heuristic solution for eq. 2.

---

**INPUT:**  $S_T, \hat{S}, \mathcal{M}_{sv}, \mathcal{L}_T, f_{s,v}^{max}, \mathcal{E}_{c,v}^{idle}, \mathcal{E}_{max}, \mathcal{T}$

**OUTPUT:**  $\mathcal{E}_T, \hat{S}(t), \mathcal{M}(t)$

---

```

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:                                      $\triangleright$  Update set of servers  $\hat{S}(t)$  with 0 penalty;
7:       Find the minimum  $\mathcal{E}_{c,v}^{idle}(t)$  in  $\hat{S}(t)$ ;
8:       if ( $flag(t) == 1$ ) then
9:                                      $\triangleright$  decrease the penalty (-1) in all the servers with  $penalty \geq 1$ ;
10:      end if
11:                                      $\triangleright$  add new VM into the  $\hat{S}(t)$ ;
12:    end if
13:  end for
    Update  $\hat{S}(t)$  and  $\mathcal{M}(t)$  through  $\mathcal{VM}(s; v)$ ;
    Calculate  $\mathcal{E}(t)$  as in (1);
14: end for
15: return  $\mathcal{E}_T, \hat{S}(t), \mathcal{M}(t)$ 

```

---

154 In order to resolve the problem, we can consider the problem as a *bin packing penalty-aware*  
 155 problem. **Algorithm 1** presents the pseudocode of the proposed solution. Indeed, in turn, for each  
 156 incoming workload  $\mathcal{L}_T(t)$  we calculate how many VMs can be allocated for  $t$ -th slot. After that, we  
 157 consider servers as bins and VMs as packs that must be served distributively based on their time  
 158 limitations and frequency limitations. For each iteration  $it$ , we give a penalty to each server based on  
 159 their energy characteristics (idle energy, maximum energy, maximum frequency) and look for the best  
 160 servers to be allocated. The penalty is considered in order to decrease the fatality cases (increasing  
 161 the energy consumptions exponentially). It means, the server will be punished and banned to be  
 162 used again in next  $it$ . When the freezing iterations  $n$ -th for the server passed, the server will back to  
 163 the list of available servers for the next  $it$ . Note that, server  $S$  can serve VMs when it does not pass  
 164 the maximum number of VMs  $\mathcal{VM}_{max}(s)$ . Otherwise, server  $S$  puts out from the list of processing  
 165 servers. The maximum interactions is defined  $\mathcal{M}_{s,v}$ . The process is done for each time slot  $t$  until all

the incoming workloads will be served. At last, we calculate the energy consumption components of the optimization problem for each slot  $t$ .

#### 4. Performance Evaluation

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

##### 4.1. Test Scenario

In order to guarantee the accuracy and timeliness of information transmission of the VCs. It is necessary to analyze and act on 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 thousands or hundreds of thousands of FNs might send periodic summaries of data grid to the Cloud for historical analysis and storage.

In order to confirm the correctness of the proposed infrastructure in Fig. 1, we require an evaluation for exploring several resource management and scheduling techniques such as operator, application and task placement and consolidation. We make some evaluations of resource management policies applicable to Fog environment with respect to their impact on latency, network usage, and energy consumption. In this framework, the VCs publish data to SVF, applications run on Fog devices (FN) to process data coming from VCs. Indeed, it is important to test the solution with real-world input workload and compare it with the corresponding ones of the recent FDC-based techniques. So in the next subsection, we will move to this issue and we will specify it with several simulated testbed cases.

##### 4.2. Test Result

In this subsection, we aim to test our scheduler on 100 incoming data stream ( $T = 100$ ) for Fast Ethernet communication channel and calculate the on-line instantaneous overall energy consumption and energy savings that is shown in Fig. 4. From this figure, we conclude that the total energy cost in changing by the time and even it has effect on its correlated saving energy per time.

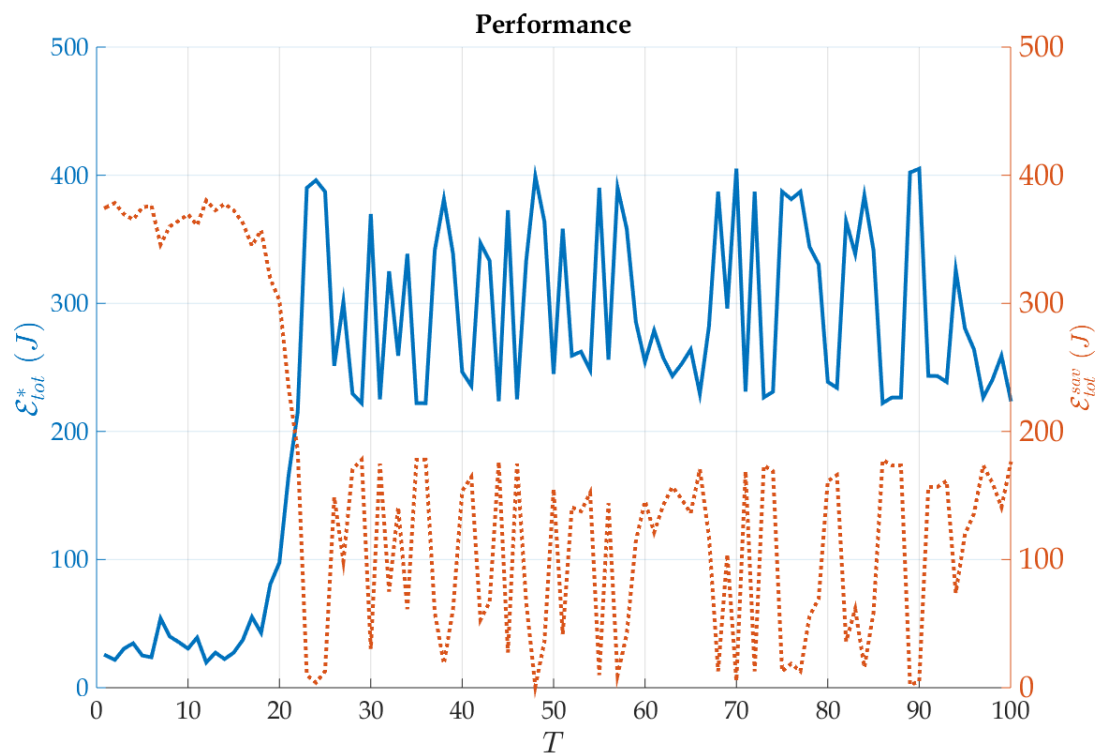
On the other hand, in order to evaluate the instantaneous energy consumption for 1000 slots using two different scheme of the communication network (Fast and Giga) we evaluate the energy consumptions of Fast/Giga Ethernet of SVF scenario. The results of these comparisons are presented in Figs. 5. We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

Finally, in the last test scenario, the plots of Fig. 6a report the time-behaviours of the numbers of VMs and physical servers that are dynamically turned ON by our algorithm scheduler under the real-world workload of Fig. 2. A comparison of the plots of Fig. 6a with the arrival trace of Fig. 2 supports the conclusion that the consolidation action performed by the proposed algorithm is capable to track the abrupt (and unpredicted) changes of the input workload with a delay of 1 slot period.

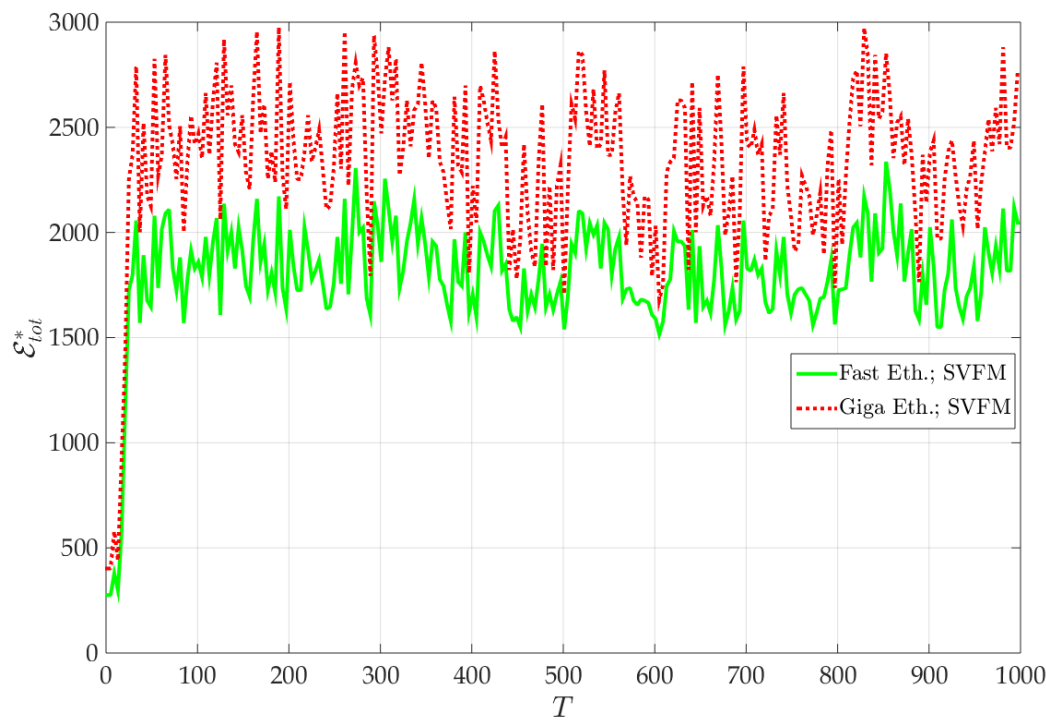
#### 5. Conclusion and Future Developments

The ability of Fog Computing is never-ending. It has the potential to solve many problems in different fields that are still unexplored such as context-aware data produce/transfer in vehicular networks. Fog computing over the vehicular networks is a platform that offers real-time services to the clients. In this paper, we study one of the most significant issues of FC which aims to focus on the Fog data center data streams which are requested from the vehicular clients. We proposed a holistic VFC architecture, focus on the FDC data streams and formulate the problem analytically and resolve the presented resource allocation and scheduling algorithm over a simplified cases study called StreamVehicularFog (SVF). In this regard, the SVF simulations in both small and large scales confirmed the considerable reduction on the average energy cost that can be achieved by scheduling in mobile edge fog of SVF. We test the presented case study with various test scenarios and confirm the energy efficiencies. The energy-efficient adaptive management of the delay-vs.-throughput trade off of the SVF TCP/IP mobile connections becomes an additional topic for further research.

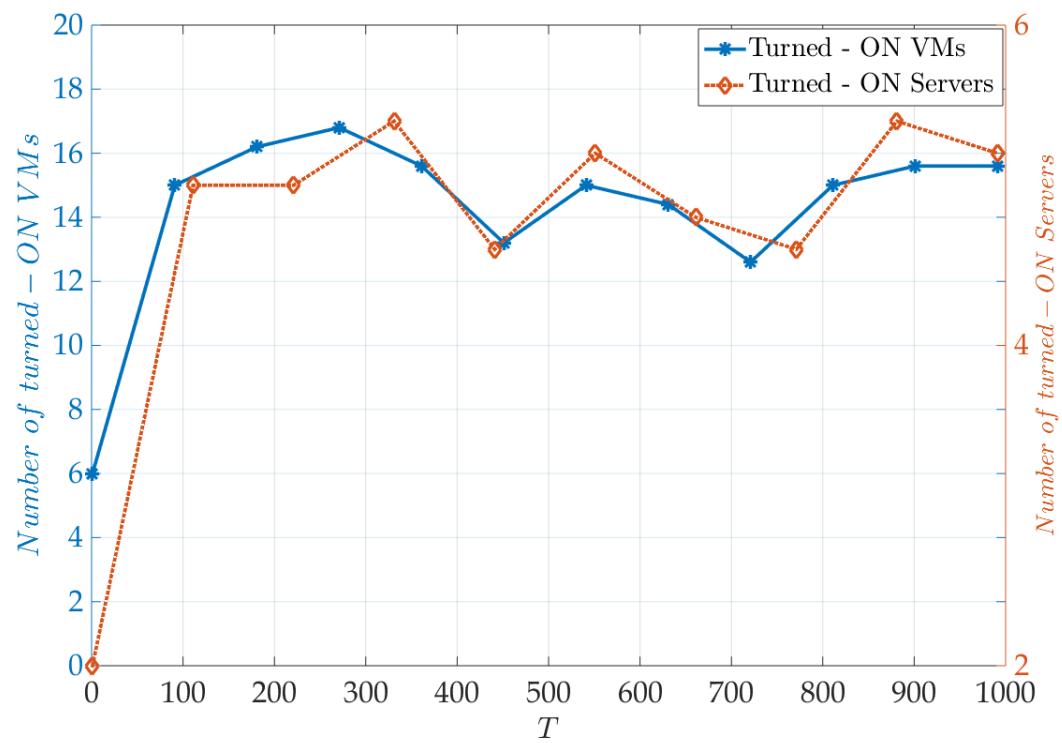




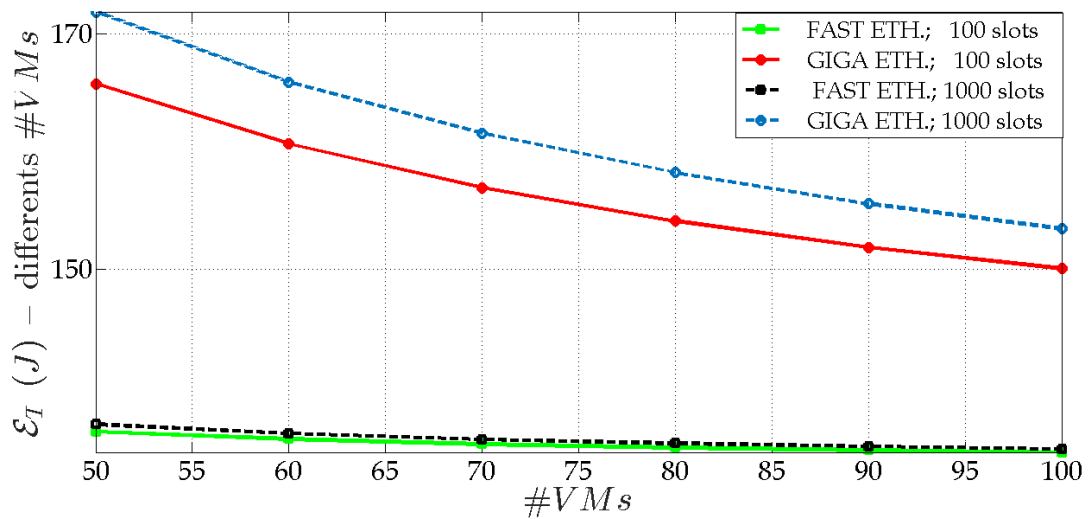
**Figure 4.** Energy saving of SVF (dashed plot) and energy consumption of the SVF (continuous plot) for the Fast Ethernet LAN for  $T = 100$ .



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



(a)



(b)

**Figure 6.** Fig. 6a: Behaviour time of number of turned-ON VMs and physical servers, and Fig. 6b: Average total energy in  $T = 1000$ , respectively.

## 6. Acknowledgement

This work has been partially supported by the PRIN2015 project with grant number 2015YPXH4W\_004: "A green adaptive FC and networking architecture (GAUCHO)", funded by the Italian MIUR and by the project "Vehicular Fog energy efficient QoS mining and dissemination of multimedia Big Data streams (V-Fog)" funded by Sapienza University of Rome.

## 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 Communications Surveys & Tutorials* **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 preprint arXiv:1710.01801* **2017**.
3. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the internet of things. Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012, pp. 13–16.
4. Büsching, F.; Schildt, S.; Wolf, L. DroidCluster: Towards Smartphone Cluster Computing—The Streets are Paved with Potential Computer Clusters. 2012 32nd International Conference on Distributed Computing Systems Workshops. IEEE, 2012, pp. 114–117.
5. Bitam, S.; Mellouk, A.; Zeadally, S. VANET-cloud: a generic cloud computing model for vehicular Ad Hoc networks. *IEEE Wireless Communications* **2015**, *22*, 96–102.
6. Mtibaa, A.; Fahim, A.; Harras, K.A.; Ammar, M.H. Towards resource sharing in mobile device clouds: Power balancing across mobile devices. ACM SIGCOMM Computer Communication Review. ACM, 2013, Vol. 43, pp. 51–56.
7. Chen, M.; Zhang, Y.; Li, Y.; Mao, S.; Leung, V.C. EMC: emotion-aware mobile cloud computing in 5G. *IEEE Network* **2015**, *29*, 32–38.
8. Skala, K.; Davidovic, D.; Afgan, E.; Sovic, I.; Sojat, Z. Scalable distributed computing hierarchy: Cloud, fog and dew computing. *Open Journal of Cloud Computing (OJCC)* **2015**, *2*, 16–24.
9. Vaquero, L.M.; Roderio-Merino, L. Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Computer Communication Review* **2014**, *44*, 27–32.
10. 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. Proceedings of the ASE BigData & SocialInformatics 2015. ACM, 2015, p. 28.
11. Aazam, M.; Huh, E.N. Fog computing and smart gateway based communication for cloud of things. Future Internet of Things and Cloud (FiCloud), 2014 International Conference on. IEEE, 2014, pp. 464–470.
12. 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. IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE, 2014, pp. 2355–2363.
13. Urgaonkar, B.; Pacifici, G.; Shenoy, P.; Spreitzer, M.; Tantawi, A. Analytic modeling of multitier internet applications. *ACM Transactions on the Web (TWEB)* **2007**, *1*, 2.