System modelling and performance evaluation of a three-tier Cloud of Things

Wei Li\textsuperscript{a,}\textsuperscript{*}, Igor Santos\textsuperscript{b}, Flavia C. Delicato\textsuperscript{b}, Paulo F. Pires\textsuperscript{b}, Luci Pirmez\textsuperscript{b}, Wei Wei\textsuperscript{c}, Houbing Song\textsuperscript{d}, Albert Zomaya\textsuperscript{a}, Samee Khan\textsuperscript{e}

\textsuperscript{a} Centre for Distributed and High Performance Computing, School of Information Technologies, The University of Sydney, NSW 2006, Australia
\textsuperscript{b} DCC/IM, Universidade Federal do Rio de Janeiro, RJ 20001-970, Brazil
\textsuperscript{c} School of Computer and Engineering, Xi’an University of Technology, Xi’an, China
\textsuperscript{d} Department of Electrical and Computer Engineering, West Virginia University, Montgomery, WV 25136, USA
\textsuperscript{e} Faculty of Electrical and Computer Engineering, North Dakota State University, ND 58102, USA

ARTICLE INFO

Article history:
Received 29 January 2016
Received in revised form
25 April 2016
Accepted 19 June 2016
Available online xxxx

Keywords:
Fog computing
Internet of Things
Cloud of Things
System modelling
Performance evaluation

ABSTRACT

The emergent paradigm of fog computing advocates that the computational resources can be extended to the edge of the network, so that the transmission latency and bandwidth burden caused by cloud computing can be effectively reduced. Moreover, fog computing can support and facilitate some kinds of applications that do not cope well with some features of cloud computing, for instance, applications that require low and predictable latency, and geographically distributed applications. However, fog computing is not a substitute but instead a powerful complement to the cloud computing. This paper focuses on studying the interplay and cooperation between the edge (fog) and the core (cloud) in the context of the Internet of Things (IoT). We first propose a three-tier system architecture and mathematically characterize each tier in terms of energy consumption and latency. After that, simulations are performed to evaluate the system performance with and without the fog involvement. The simulation results show that the three-tier system outperforms the two-tier system in terms of the assessed metrics.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

In the past few years, we have witnessed a wide adoption of Internet of things (IoT) [1] around the world, which has become one of the most promising technologies to enable ubiquitous and pervasive computing scenarios in the future. IoT can be considered as the interconnection of physical objects (also known as things) that contain embedded technology to sense and communicate or to interact with the external environment or other physical objects. The ever-increasing number of IoT devices will inevitably result in generating enormous amount of data, which has to be processed, stored and properly accessed by end users and/or client applications. In addition, the data from the physical objects will need to be combined with other computing resources available in the web to produce value-added information for the user. Cloud computing [2] has long been recognized as a service-oriented paradigm for big data storage and analytics. The combination of IoT and cloud computing brought about a new paradigm of pervasive and ubiquitous computing, called Cloud of Things (CoT) [3]. CoT is a recently proposed approach to integrate heterogeneous resources of networked physical objects with virtually unlimited computing and storage capacities from public/private clouds in order to create infrastructures capable of delivering new generation services.

A CoT is composed of virtual things built on top of the networked physical objects and provides on-demand provision of sensing, computational and actuation services, allowing end users to pull the data from the real things by using service-oriented systems, anywhere in the world. It thus allows the global sharing of IoT resources that are naturally distributed, elastic resource provisioning from things and cloud platforms and flexible interaction with customers [4]. Moreover, with the help of virtualization technique and software-as-a-service model, the developers can make effective use of virtual things according to their requirements to build up applications without the need of handling the low-level underlying hardware details. The current studies [4,5] generally model the architecture of CoT as a two-tier system, cloud and physical entity, as shown schematically in Fig. 1.

http://dx.doi.org/10.1016/j.future.2016.06.019
0167-739X/© 2016 Elsevier B.V. All rights reserved.

Please cite this article in press as: W. Li et al., System modelling and performance evaluation of a three-tier Cloud of Things, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.06.019
The development of cloud computing over the last decade places a premium on the proliferation of large data centres [2], so that data and applications gradually move into the centralized data centres, where the majority of computation occurs. While this approach has many benefits, it can also be very costly in terms of communication and energy consumption. When integrating the traditional cloud computing systems with physical objects, especially when such objects are located in close proximity of each other or of the end users, it can be highly inefficient in terms of time to transmit every bit of sensed data from those things to a distant data centre for processing, and then transmit the result back to the end users or to the individual things for actuations. Since this approach could generate heavy communication overhead for the network as well as long latency for the applications, the users of time-constraint IoT applications are highly likely to experience a poor-quality service.

In IoT, a Physical Entity denotes any physical object of interest (for monitoring and/or actuation purposes). Such Physical Entities are endowed with devices, which are equipped with sensing and/or actuating units that are responsible for collecting the physical variables of interest, or for acting over the Physical Entity to change their state. A Physical Entity can generate different types of sensing data depending on the devices instrumenting it. As shown in Fig. 1, if two devices from different authorities are used to monitor the same Physical Entity at the same time, the same result could be collected but sent to different clouds for processing or storage. This may cause extra computation resource usage, inefficient data utilization and unnecessary data replication. Besides that, some other issues, such as: improving the duration of battery operated devices (or their operation autonomy, a natural consequence of less changes/recharges of batteries), collect fresh and accurate context information and support mobility, impose additional challenges to the existing CoT systems. To address such issues, a new architecture design tailored to the distinctive features and characteristics of CoT is needed.

Recently, there has been a noticeable trend on introducing the concept of edge computing [6] into CoT in order to push the frontier of applications, data and services away from the centralized nodes (the “core”) to the periphery of the network (the “edge”). The edge computing paradigm does not refer to the purely decentralized systems or P2P systems. It is a federation of resource-poor devices, e.g. sensors, RFID (radio frequency identification), human-controlled devices, e.g. smartphone, tablets, stable networking devices, e.g. switches, routers and resource-rich machines, e.g. cloudlets [7–9]. With these edge devices, several types of data processing can happen at the edge of the network, such as to pre-process the sensed data and trigger alerts and actuation requests locally based on predefined criteria. Additionally, such a system may leverage local connectivity technologies to enable direct, more efficient and decentralized communication between things. As a result, the edge computing paradigm still retain core advantages of using clouds as a supporting infrastructure, but will put
back some decision making and data processing to the edge and open up new possibilities for enabling distributed intelligences in the Internet of things. A recently used term in this area is fog computing [10], which was popularized by Cisco. Fog computing refers to a computing paradigm that extends cloud computing and some of its provided services to the edge of the network. In contrast to cloud, fog spans more condensed geographic locations in a much denser way. Some desired capabilities that are originally located at cloud, such as computation and storage can be gradually moved back to the network edge and provided to the things. As a direct result, fog computing reduces service latency, improves QoS (Quality of Service) and provides better overall experience to end users. These features are particularly useful for several real-time applications, such as vehicular networks, smart grid, pipeline monitoring, connected rail, smart traffic light systems, wind farms, closed loop control of industrial systems, and applications in the oil and gas sector. As a long-term goal, fog computing can slowly change the current centralized computation paradigm of cloud computing to the distributed computation paradigm again, so that the control, data and intelligence of IoT systems will no longer be only available at the cloud.

Alongside to the emergence of the concept of edge computing, the ETSI (European Telecommunications Standards Institute) launched an industry specification for Mobile Edge Computing (MEC) [11–14] in September 2014. This group is developing a system architecture and standardizing a set of APIs for mobile edge computing. Fog and MEC both address similar issues and are about moving processing from the cloud to the edge of the network, supporting node mobility, and providing low latency, real-time interactions and location awareness. However, there are some relevant differences between Fog and MEC. Among such differences, fog computing paradigm stresses the need for efficient communication between edge nodes while MEC does not. IoT and Wireless Sensor and Actuator Networks (WSAN) were major driven forces that inspired fog, while MEC was mainly inspired by NOKIA Liquid Apps, Telecom use cases like content acceleration, and hardware details. By doing so, a physical thing can be treated as a real hardware or as a virtual thing that is used by a single application.

As we mentioned before, fog computing is not a replacement of cloud computing. Instead, we treat fog computing as an essential supplement of cloud computing to address the new requirements raised by several applications, particularly in the context of IoT.

To the best of our knowledge, this is the first effort towards providing a systematic framework for a fog integrated CoT and supporting distributed intelligence at the network edge in IoT. The contributions of this work are listed below:

- We present the system architecture for a fog integrated three-tier CoT system, in which the fog tier is physically placed at the edge of network so as to directly communicate with things.
- We employ a newly developed concept called sensor function virtualization (SFV) [15] as a key enabler to construct a complete abstraction of the underlying physical things that belong to different authorities, regardless the network topology and hardware details. By doing so, a physical thing can be envisioned as a provider of multiple services, rather than being treated as a real hardware or as a virtual thing that is used by a single application.
- We establish the mathematical model of our three-tier CoT system, so that we can further study the system performance, as well as the suitability of integrating fog computing into the traditional CoT, in a theoretical way.
- Based on our extensive simulation experiments and numerical results, we quantitatively reveal how the integration of fog computing can considerably affect the performance of CoT in terms of reducing the service latency and energy consumption.

The rest of the paper is organized as follows. Section 2 provides an overview of the work that has been done in the field. In Section 3, the detailed architecture of the fog-integrated CoT system is presented. The mathematical models that used to describe our proposed system, as well as the performance metrics are then introduced in Section 4. In Section 5, the performance evaluation of our three-tier CoT system is performed and a comparative study of both two-tier and three-tier CoT systems is provided. Moreover, a simple version of the proposal was implemented as a Proof-of-Concept to illustrate its usage in a real setting. Finally, the work is concluded and future work is discussed in Section 6.

2. Related work

In the last few years, IoT and cloud computing have both gained popularity as complementary technologies. The recent integration of IoT and Cloud Computing—CoT, as a promising topic, has attracted increasing attention from both research and industry. Most of existing works on CoT is driven by the purpose of using cloud to fulfill some requirements of IoT, such as: computation and storage [16]. With a substantial amount of information generated by things, IoT certainly needs intensive computation resources from cloud for online data processing, such as performing real-time data analysis, implementing scalable, collaborative and data-centric applications, and managing complex events. The data produced by IoT devices can be also used for offline data processing, including sophisticated data fusion, data prediction and data reconstruction tasks. Once the data is stored in the cloud, it can be treated as an information source, directly accessed and visualized from any place and protected by security and privacy mechanisms. In addition, the development of CoT enables new value-added services through the composition of existing cloud architecture with the things, which enable the cloud to connect with physical world, giving birth to the things as a service paradigm. There are several emerging paradigms [16] related to CoT, including things as a service [4], sensing as a service [5], sensing and actuation as a service [17], sensor event as a service [18] and sensor as a service [19]. However, due to lack of standards in this emergent field, there is no sharp distinction among the proposed acronyms, which in some cases appear to collide. Similar to the well-known federated infrastructures, e.g. grids, CoT was also motivated by utilizing the heterogeneous and dynamic computation, storage and network resources from multiple geographically distributed locations to address the needs of applications. However, as discussed in [20], grids focus on how to integrate the remote resources with their existing hardware, operating systems, local resource management, and security infrastructures. Particularly, the aspects of interoperability and security are major concerns for the Grid infrastructure as resources may come from different administrative sites, which have both global and local resource usage policies, different hardware and software configurations and platforms, and vary in both availability and capacity. To address these issues in CoT is certainly out of the scope of our paper.

Despite many benefits that can be brought by the combination of cloud and IoT, there is still a noticeable bottleneck in such kind of system, which is the communication capacity of the Internet. Although the communication capacity of Internet has been increasing by a factor of $10^{4}$ over the last 20 years, the processing power and storage density have increased by $10^{15}$ and $10^{18}$, respectively [21]. Consequently, practical limitations can arise when trying to transfer vast amounts of data from the edge of the network onto the cloud, especially for those time-constrained IoT applications. For addressing the needs of the time-constrained IoT applications, fog computing has been proposed as a distributed computing infrastructure to minimize...
the transmission distance between things and cloud. Thus, the data collected from the sensing devices can be processed in the fog layer before one can decide if the processing outcome is to be sent to the final destinations, either the cloud or back to the devices. The basic definition and characteristics of fog computing are given in [10], where the authors also identified several specific requirements of IoT, e.g. location awareness, heterogeneity, low latency, geographical distribution, mobility support, real-time interactions and the predominance access to wireless devices. In [22], the authors not just provide a comprehensive definition of fog computing, but also highlight the main challenges that will have to be addressed so that fog computing can mature and unfold its entire potential. Several potential IoT applications that need the help from fog computing are also introduced in different literatures, e.g. content delivery and caching [23], mobile big data analytics [24], vehicular fog computing networks [25] and e-health applications [26]. These works cover different aspects of fog computing, but most of them seem to study it merely from a qualitative perspective. It is worth to mention that the same concern is attracting the attention of the researchers to provide solutions from different aspects, e.g. emerging sensing and communication technologies in wireless sensor networks [27,28], mobile sink node management [29,30], task scheduling for mobile devices [31] and applications optimization of mobile cloud computing [32–34].

Only very recently, few researchers started exploring the fog computing from a quantitative point of view. The authors in [35] proposed several mathematical models of fog computing and based on these models, they analysed the power-delay tradeoff problem in the cloud–fog computing system. It is worth to mention that the system model presented in [35] adopts wired connection between devices and the fog computing subsystem, which is not well aligned to the characteristic of “the predominance access to wireless devices” mentioned in the pioneer work [10] presented by Cisco. In [36], the authors provided a detailed mathematical model to describe the characteristics of fog computing network in terms of power consumption, service latency, CO₂ emission and cost. Their work is only focused on the interactions between fog layer and cloud layer. In other words, the performance study is purely based on a cloud computing system that extends its computing capability outside of the data centre. In [37], the authors proposed a smart gateway based communication for integrating IoT with cloud computing. Data collected from IoT devices is sent to the smart gateway via either one-hop or multi-hop communication. The proposed smart gateway focused on extending traditional gateways to provide service-oriented functionalities besides the proxy and network functionalities. The IoT data is pre-processed before sending to the Cloud. However, no quantitative method is provided for measuring the system performance. In our work, we first present a three-tier CoT system architecture, including things, fog and cloud. The things are equipped with devices (sensors and actuators), accessed through set of services and can be used by multiple applications. Our work also provides the related mathematical models to evaluate its appropriateness. With such models, we can further investigate the overall performance of our proposed three-tier system and evaluate how the involvement of fog computing can affect the CoT system.

3. The system architecture of fog-integrated CoT

In this section, we present the overall architecture of our proposed three-tier CoT system and the details related to each tier.

As depicted in Fig. 2, our proposed CoT system contains three distinct physical tiers, from tier 1 to tier 3. The tier 1 is also known as things tier. In our architecture, it is the bottom tier and it is responsible for connecting the physical world to the digital world. It encompasses multiple types of Physical Entities (objects of interest), ranging from static ones, e.g. bridges and buildings to moving ones, e.g. animals and vehicles. Physical Entities are endowed with devices, which are equipped with sensing and/or actuating components that are responsible for collecting the physical phenomenon of interest (by monitoring the entities) and send the collected data back to upper layers for further processing, or for acting over the physical entities to change their state. A Physical Entity can generate different types of sensing data depending on the device instrumenting it. For instance, a bridge can produce acceleration, temperature, humidity and pressure data. Different applications are interested in different types of data. In our proposed model, each type of data is offered as a service provided by a Virtual Entity. Therefore, the concept of Virtual Entity can be defined as one or more sensing/acting services associated with a Physical Entity.

The tier 2 is the fog computing tier in our architecture. Fog nodes are the key components of this tier. Fog nodes are devices that can provide resources (processing and storage) for running services at the edge of the network [23]. The involvement of fog computing tier is one of the noticeable differences between our three-tier CoT architecture and some other existing CoT architectures [38,39]. In those previous works, although the logical three-tier architecture is suggested, the middle tier is either physically located at the cloud tier or at the things tier, which indicates these proposals are in fact a two-tier architecture design from the system point of view.

The tier 3 is the cloud tier. The key component in this tier are the data centres, which are used to provide the required computing capability and storage to the users and applications based on the pay-as-you-go or a utility-like pricing model.

As discussed earlier, the tier 1 comprises of things that are Physical Entities instrumented by sensing or actuation devices. Several characteristics in this tier are worth mentioning for better understanding our proposed system architecture. First, the things in our system could belong to different authorities. Physical Entities located at the same geographical area could form different logical groups via different mechanisms, such as self-organization protocols. Things can eventually move from one area to another. These mobile entities can be still accessed by the upper layers during the movement by using direct/indirect routing technologies without changing their ownership. In addition, different kinds of devices generally exist in a given area. These devices can form different topologies to work collaboratively and transmit the data (achieved from the monitored things) to the upper tier via one-hop or multiple-hops communication. Last, the things might be equipped with GPS units to accurately acquire their geographic location and be able to share this information among things, as well as with the upper layers. In addition to that, the approximate geographic locations of things could also be calculated by different positioning techniques with the help of their nearby access point or things that have their precise geographic location. With this information, the location-aware services that are identified as one of the main characteristics of fog computing [10] could thus be provided to the end users.

In our system, the access points are located in amid the things tier and the fog tier for transmitting the data between these two tiers. They can be classified into two types, namely single service access point and integrated services access point. The single access point can only handle the devices compliant with a specific communication protocol and any attempted request from the devices based on different types of protocols will be rejected. In contrast to the single access point, the integrated services access point can provide network access for the things/devices based on different protocols in the same area. Please note that, we assume that all the access points are located at the edge of the network and
only at one hop wireless transmission distance from the things tier. After received by the access points, the data will then be passed to the fog nodes (tier 2) for further processing.

As proposed by [23], the fog computing architecture can be further divided into two parts, (i) the fog abstraction layer and (ii) the fog orchestration layer. The fog abstraction layer is responsible for managing different fog nodes and virtualising those nodes. By doing this, the computation and storage can be extracted from various kinds of ICT devices, such as: routers, switches, computing machines and access points. Although Cisco claims that fog nodes are typically but not exclusively located at the edge of network [10], the fog nodes placed close to things can significantly reduce the latency that might be generated from the data transmission over Internet as well as provide real-time interactions for the time-constrained IoT applications. According to this setting, in our system the fog nodes are naturally bundled in regional fog groups so that the data transmission delay can be controlled in a predictable and lower value compared to the delay occurred in the global Internet. The resources of each regional fog group are normally used to process the data from the access points located at the same area. The fog orchestration layer comprises of a small software agent—often called foglet [23], which monitors the state of the fog nodes, a distributed database to account for scalability and fault tolerance. Besides that, a service orchestration module at the same layer is responsible for policy-based routing of application requests. Besides that, the potential federation among regional fog groups and the collaboration between the fog tier and the cloud tier are also handled at this layer. For example, when the sensing data produced by things is passed from the lower layers, it will be properly analysed to decide whether it needs to be transmitted to the cloud. If the received data is used for the aggregation with the collected data from distant things or if the data needs to be analysed along with historical data, it will then be forwarded to the cloud. Otherwise, the data can be processed by
the fog nodes in the regional fog groups. Apart from these desired features of the fog computing, our fog tier provides one more feature to address the needs of CoT. As mentioned before, Physical Entities (PE) provide their services by means of Virtual Entities (VE). Such VEs are normally located at the cloud, which does not address well the need of localization. Instead, we not only provide the VEs at the cloud tier, but also provide them at the fog tier. With the placement of VEs at the fog tier, the localization information will be utilized more efficiently and easily than in the cloud, so that leading to better results of low latency and context awareness. In addition, each VE can be comprised of different functions from physical things located at different areas and belonged to different authorities. Please note that, the virtualization models, detailing how a given virtual entity is built from the composition of one or more VEs as well as the cooperation model describing the interaction among entities of the different tiers are out of the scope of this paper and left as future work.

In our model, the cloud tier provides global centralization of available resources and is responsible for data-intensive computation and permanent storage of huge, valuable data chunks within its data centres. With the localized VE information provided at the fog tier, the servers in the cloud can better utilize the up-to-date information to meet the requirements of applications and end users. Moreover, some operations supposed to be performed at the cloud tier are now moved to the fog tier due to fog nodes have relatively rich resources and closer physical distance to things to process these requests, especially for those requests that require real-time processing (from milliseconds to sub seconds). Compared to the two-tier model, the cloud tier in our model is no longer to be bothered by every single request so that the resources can be reserved for other usage. In other words, the involvement of fog tier enables the cloud tier to be accessed and utilized in a more efficient manner.

4. Modelling

This section is divided into four subsections, where we detail the models which are used in our proposed three-tier CoT system, including: application model, system model, energy model and time-delay model.

4.1. Application model

In our proposal, an application is defined as a 4-tuple \((V, E, G, D)\), where \(V\) denotes the tasks of an application, where the number of the tasks ranges from \([1, n]\), \(n \in \mathbb{Z}^+\). Each task denotes a non-communication module, such as a computing function or a sensing function in a program. In other words, each task represents a required service that is potentially provided by the three-tier CoT. \(E\) is called a directed graph and it denotes the distinguished direction between a pair of tasks \((u, v), u, v \in V\), where \(E\) represents the precedence in between task \(u\) and \(v\). \(G\) is the geographic area, determined by four geographic coordinates that define the boundary of an area which the application is required to action. \(D\) is used to specify the expected deadline of the application. In the context of our work, task and service are strongly correlated. A service indicates the action that can be performed by a component of the 3-tier CoT system so as to be requested for accomplishing the related task. A task \(v \in V\) may have the following properties: (service_type, workload). The service_type element denotes the type of the service a task requests. Each task is associated with one and only one service type (for instance, sensing, processing, communication, actuation etc.). The workload element represents the duration of a task, which means how long it is going to be processed once it gets the service from the CoT. The CoT applications considered in this paper consist of at least one task; thus an application may require a composition of services. Specifically, each of these tasks is the finest-grained and non-divisible element to constitute the CoT application. In this paper, we assume that tasks are dependent from each other. The dependence that we are addressing in this paper is the data dependence, which reflects the dependence relations between tasks caused by data transfer. In other words, a task \(u\) is the predecessor of task \(v\) and correspondingly \(v\) is the successor of \(u\), if and only if edge \(e_{uv} \in E, u, v \in V\). We also assume that once a task is initiated, it cannot be interrupted until its completion (non-preemptive task).

4.2. System model

In the things tier, the lowest tier as depicted in Fig. 2, we assume that there are a total of \(k_{area}\) areas denoted by the set \(O = \{O_1, O_2, \ldots, O_{k_{area}}\}\). Each given area \(O_i, O_i \in O\) contains Physical Entities \(PE_i^m = \{PE_i^1, PE_i^2, \ldots, PE_i^m\}, m \in \mathbb{Z}^+\) and each Physical Entity \(PE_i^m\) is interpreted to be an element from the set \(T = \{T_1, T_2, \ldots, T_n\}\). PEs can be dynamically registered into or deregistered from the three-tier CoT system. Each given Physical Entity \(T_i\) is associated to one or more Virtual Entity \(VE = \{VE_1, VE_2, \ldots, VE_n\}\). \(n \in \mathbb{Z}^+\) is responsible for providing one or more services \(T_i = \{T_i^1, T_i^2, \ldots, T_i^n\}\). \(T_i \in T\) depending on the capabilities of the devices instrumenting the PE.

We assume that each Physical Entity \(T_i, T_i \in T\) within a given area \(O_i, O_i \in O\) always has a valid communication path to reach its upper tier via the access points. For the sake of simplicity, communication interference is not considered in this paper.

In order to examine the service availability of a VE, we introduce a judging function that indicates whether a VE can provide the required service at the specified area. The judging function is shown below:

\[
A(s, x, y, O_i, l) = \begin{cases} 
1, & \text{if a VE is available} \\
0, & \text{if a VE is not available} 
\end{cases}
\]

where \(s\) is the service that an application requires, \(x\) and \(y\) are the location information for the PE associated to the VE, \(O_i\) is the given area at the thing tier and \(l\) is the index of the PE in the area \(O_i\).

In the fog tier, there are two types of devices considered in our study, access points and fog nodes. The data generated by all the constituent things deployed into the same area will be firstly transmitted to the access points placed in that area. For a given area \(O_i, O_i \in O\), there exists a non-empty set of access points \(AP_i^m = \{AP_i^1, AP_i^2, \ldots, AP_i^m\}\) that provide network access and data transmission services for the things located at that area. For each access point \(AP_i^m, O_i \in O, AP_i \in AP_i^m\), we assume a high frequency antenna (e.g. 60 GHz) is used for providing high directivity with high data rate (4–15 Gbps) [40] so that the data can be collected from a number of things simultaneously. Once the data is received by the access point, it will be forwarded to fog nodes within the regional fog via wired communication. A specific regional fog might need to process the data collected from multiple things that are located in its coverage area. We assume that, at a given time \(t\), the number of regional fogs \(k_{fog}\) is not greater than the number of areas \(k_{area}\) in the system, that is \(k_{fog} \leq k_{area}\). When the data reaches the fog nodes, a decision is made on whether the data will be processed at the fog tier or forwarded to the upper tier based on the deadline allocated to the application. If an application demands real-time processing, it will then be processed within the fog nodes in the related regional fog groups rather than forwarded to the (distant) data centres in the cloud tier. At the meantime, the intermediate data that is useful for

\[
O = \{O_1, O_2, \ldots, O_{k_{area}}\}.
\]
future use would also be stored locally in the fog tier. However, if an application requires intervention of the cloud computing layer for data analysis based on historical data-sets and for long-term storage, then the generated data is redirected to the upper tier.

As we mentioned before, the available functions of each Physical Entity are presented as services via sensor function virtualization. The constructed VEs are located at both fog and cloud tiers. For a given VE in the fog tier, there exists a limit on the number of working machines purchased from that centre. Let \( f_{\text{min}} \) and \( f_{\text{max}} \) denote the lower and upper bound on the data centre working frequency, respectively:

\[
f_{\text{min}} \leq f_j \leq f_{\text{max}} \quad \forall j \in \mathbb{M}.
\]

In addition, for the cloud server \( j \), the number of working machines \( n_j \) has an upper bound \( N_j \). Thus, for the integer variable \( n_j \), we have

\[
n_j \in \{0, 1, 2, \ldots, N_j\} \quad \forall j \in \mathbb{M}.
\]

### 4.3. Energy consumption models

In order to analyse the energy consumption in the proposed three-tier CoT system, we need to identify the main sources of energy consumption at each tier. In the remainder of this subsection, we analyse the energy consumption of each element of such tiers.

In the first tier, the main source of energy consumption is the device, which can be composed of four major components: sensing components, computation component, communication component and actuation component. Inspired by the work \cite{41}, the below formulas used to represent the energy consumption of things are adopted and modified from it. The main sources of power consumption at the sensing components are: signal sampling and conversion of physical signals to electrical signals, signal conditioning, and analog to digital conversion (ADC). Let \( I_{\text{sens}} \) be the total current required for sensing activity and \( E_{\text{sense}} \) be the time duration for sensing component to collect data from the PE.

![Image](https://example.com/image.png)

**Table 1**

<table>
<thead>
<tr>
<th>CoT hardware parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total simulation time (duration)</td>
<td>1 day</td>
</tr>
<tr>
<td>Total sensing area ( G ) (square)</td>
<td>200 m ( \times ) 200 m</td>
</tr>
<tr>
<td>Access points coverage radius</td>
<td>20 m</td>
</tr>
<tr>
<td>Total number of fog nodes ( (= ) number of areas ( O ))</td>
<td>5 nodes</td>
</tr>
<tr>
<td>Number of physical entities per area ( O )</td>
<td>20 nodes</td>
</tr>
<tr>
<td>Total number of physical entities in simulation</td>
<td>100 nodes</td>
</tr>
<tr>
<td>Number of homogeneous processing cores of each fog node</td>
<td>4</td>
</tr>
<tr>
<td>Number of homogeneous processing cores of cloud node</td>
<td>20</td>
</tr>
<tr>
<td>Standard core speed of fog nodes/cloud node</td>
<td>( 3 \times 10^6 ) cycles/s</td>
</tr>
<tr>
<td>Default transmission delay at fog level (delay models)</td>
<td>0.2</td>
</tr>
<tr>
<td>( E_{\text{trans}} )</td>
<td>50 nJ/b</td>
</tr>
<tr>
<td>( f_{\text{sup}} )</td>
<td>10 pJ/b/m²</td>
</tr>
<tr>
<td>Default x of radio</td>
<td>2</td>
</tr>
<tr>
<td>( V_c )</td>
<td>26 mV</td>
</tr>
<tr>
<td>( C )</td>
<td>0.67 nF</td>
</tr>
<tr>
<td>( I_b )</td>
<td>1.196 mA</td>
</tr>
<tr>
<td>( n )</td>
<td>2.126</td>
</tr>
<tr>
<td>( K )</td>
<td>239.28 MHz/V</td>
</tr>
<tr>
<td>( V_{\text{dd}} ) and ( V_{\text{sup}} )</td>
<td>1 V</td>
</tr>
<tr>
<td>( c )</td>
<td>0.5 V</td>
</tr>
<tr>
<td>Default energy for actuation and number of actuations</td>
<td>( 0.2 ) J, ( 1 ) act</td>
</tr>
<tr>
<td>Default ( I_{\text{act}} )</td>
<td>1 mA</td>
</tr>
<tr>
<td>Default static power of fog nodes</td>
<td>40 W</td>
</tr>
<tr>
<td>Default maximum power of fog nodes</td>
<td>400 W</td>
</tr>
<tr>
<td>Default LDR of fog nodes</td>
<td>0</td>
</tr>
<tr>
<td>Default static power of cloud node</td>
<td>100 W</td>
</tr>
<tr>
<td>Default maximum power of cloud node</td>
<td>400 W</td>
</tr>
<tr>
<td>Default LDR of cloud node</td>
<td>0.0</td>
</tr>
<tr>
<td>Default ( c_{\text{act}} )</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Table 2**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Acronym</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soft deadline missing ratio</td>
<td>S-MR</td>
<td>Q1</td>
</tr>
<tr>
<td>Average distance to soft deadline</td>
<td>S-AD</td>
<td></td>
</tr>
<tr>
<td>Hard deadline missing ratio</td>
<td>H-MR</td>
<td></td>
</tr>
<tr>
<td>Average distance to hard deadline</td>
<td>H-AD</td>
<td></td>
</tr>
<tr>
<td>Total energy spent by things tier</td>
<td>TES</td>
<td></td>
</tr>
<tr>
<td>Total energy spent by fog tier</td>
<td>TESf</td>
<td></td>
</tr>
<tr>
<td>Total energy spent by cloud tier</td>
<td>TESc</td>
<td></td>
</tr>
<tr>
<td>Total energy spent by the whole system</td>
<td>TESs</td>
<td></td>
</tr>
<tr>
<td>Average energy spent by a PE in things tier</td>
<td>AES</td>
<td></td>
</tr>
<tr>
<td>Average energy spent by a node in fog tier</td>
<td>AESf</td>
<td></td>
</tr>
<tr>
<td>Maximum energy spent by a PE in things tier</td>
<td>MES</td>
<td></td>
</tr>
<tr>
<td>Lifetime of a PE</td>
<td>LPE</td>
<td></td>
</tr>
</tbody>
</table>

Please cite this article in press as: W. Li et al., System modelling and performance evaluation of a three-tier Cloud of Things, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.06.019
Fig. 3. Results of (a) S-MR, (b) S-AD, (c) H-MR and (d) H-AD metrics with different number of applications.

The total energy consumption $E_{\text{sens}}$ is evaluated by

$$E_{\text{sens}}(b) = bV_{\text{sup}}I_{\text{sens}}\xi_{\text{sens}}$$  \hspace{1cm} (5)

where $b$ is number of bit packet is collected by the sensing activity, $V_{\text{sup}}$ is the supply voltage, $I_{\text{sens}}$ is the total current required for sensing activity and $\xi_{\text{sens}}$ be the time duration for sensing component is collecting data from the PE. Communication among neighbouring devices is enabled by a wireless radio.

Energy dissipation of a device can be attributed to transmitting and receiving data. Considering the fact that a device consumes different energy when it sends or receives the same amount of data over the same distance within its communication range, we distinguish them as $E_{\text{tx}}(l, d)$ and $E_{\text{rx}}(l)$ where $l$ represents the size of the transmission data and $d$ represents the distance between two communicating devices. The equations are listed below:

$$E_{\text{tx}}(l, d) = E_{\text{elec}} \times l + \varepsilon_{\text{amp}} \times l \times d^x$$  \hspace{1cm} (6)

$$E_{\text{rx}}(l) = E_{\text{elec}} \times l$$  \hspace{1cm} (7)

where $E_{\text{elec}}$ and $\varepsilon_{\text{amp}}$ are hardware-related parameters, which denotes radio energy dissipation and transmission amplifier energy dissipation, respectively. The value of $x$ in Eq. (6) can be varied from 2 to 4; in our work we adopted the value of 2 for the sake of easy analysis. The energy consumption for processing a task on a given device, is attributed to two components: energy loss from switching, $E_{\text{switch}}$, and energy loss due to leakage current, $E_{\text{leak}}$. Thus, the total energy consumption by the device used for data processing/aggregation for each unit of data of a task, $E_{\text{comp}}$, is given by

$$E_{\text{comp}}(V_{\text{dd}}, f) = NCV_{\text{dd}}^2 + V_{\text{dd}} \left( \frac{V_{\text{dd}}}{V_{\text{T}}} \right) \left( \frac{N}{f} \right)$$  \hspace{1cm} (8)

$$f \approx K(V_{\text{dd}} - c)$$  \hspace{1cm} (9)

where $V_{\text{T}}$ denotes the thermal voltage and $C$, $I_b$, $n$, $K$, and $c$ are processor-dependent parameters. The last part of energy consumption for a device is actuation, $E_{\text{actu}}$. However, it is hard to estimate in general because this is highly dependent on the specific actuation task (from simply blinking a led to more sophisticated operation such as moving a robot arm). Total energy consumption for actuation is $E_{\text{actu}}N_{\text{act}}$ where $N_{\text{act}}$ is the number of actuations. For example if we use temperature sensors to drive a fan that needs two motors, there can be a command to switch on the two motors when temperature is beyond some value and in that case $N_{\text{act}} = 2$. Overall, the energy consumption in the things tier can be represented as below:

$$E_{\text{Things}} = E_{\text{sens}} + E_{\text{tx}} + E_{\text{rx}} + E_{\text{comp}} + E_{\text{actu}}.$$  \hspace{1cm} (10)

In the fog tier, the main source of energy consumption is the fog devices. Energy consumption in a fog device can be divided into two parts: static and dynamic power consumption. Static power $E_s$ is consumed even if the machine is idle, while the dynamic power $E_d$ is proportional to the resource utilization within the host.

$$E_{\text{fog}} = E_s + E_d = E_s + (E_{\text{max}} - E_s) \times U$$  \hspace{1cm} (11)

where $U$ is the utilization level of the host and $E_{\text{max}}$ indicates nominal power as the maximum power device can dissipate. In the above equation, we assume a linear correlation among the utilization level and the power drain in the host, which is known as hypothetical linear power model [42]. This model is an ideal power model that let the value of Linear Deviation Ratio (LDR) to be equal to 1. However, in real systems, the LDR is not equal to 1. LDR is a metric defined as maximum difference of the actual
power consumption and hypothetical linear power model over the hypothetical linear power model as in below.

\[
LDR = \frac{E(U) - [E_s + E_{max} - E_i] \times U}{E_s + (E_{max} - E_i) \times U}.
\]  

(12)

\(E_s\) for hosts in the large data centre is above 100 W. The regional fogs can somehow compare to the data centre due to they also contain highly virtualized devices [10], but in a much smaller scale. The static energy consumption of fog devices is thus trivial comparing with the data centre hosts. By sharing part of their resources, the fog devices are contributing in the regional fog, e.g. data forwarding and data processing. However, idle energy consumed in regional fogs are smaller than in the data centre hosts because the higher effective utilization of energy of fog devices.

In the cloud tier, the main source of energy consumption is the hosts and the cooling system. Energy consumption for task execution in the cloud includes computing energy consumption and cooling energy consumption. Cooling energy consumption is the energy consumed to keep the temperature in a data centre constant. Let \(E_{\text{CloudComp}}(l)\) denote the computing energy consumption for executing a task with workload \(l\) and \(E_{\text{cool}}(l)\) the cooling energy consumption, respectively. As shown in [43], \(E_{\text{cool}}(l)\) is highly relative to \(E_{\text{CloudComp}}(l)\), and can be calculated as

\[
E_{\text{cool}}(l) = \frac{E_{\text{CloudComp}}(l)}{\text{CoP}(TE_{\text{sup}})}.
\]  

(13)

where \(\text{CoP}(TE_{\text{sup}})\) is the coefficient of performance for the cooling system to supply cool air at temperature \(TE_{\text{sup}}\). The calculation of \(E_{\text{CloudComp}}(l)\) is the same as \(E_{\text{fog}}\). The total energy consumption for executing a task in the cloud can be represented as below:

\[
E_{\text{Cloud}}(l) = E_{\text{CloudComp}}(l) + E_{\text{cool}}(l) = \left(1 + \frac{1}{\text{CoP}(TE_{\text{sup}})}\right) \times E_{\text{CloudComp}}(l).
\]  

(14)

4.4. Time-latency models

The time-latency of completing an application in the CoT can be roughly divided into two types: processing latency and transmission latency. The processing latency is the response time for all the tasks of an application being processed in the system. The transmission delay is the communication time of sending a unit of data from the source to the destination. As mentioned before, the processing latency and the transmission latency can be easily computed from the workload of the given task and the communication overhead of the incoming and outgoing edges associated with that task. The processing speed at each tier is different, and the below relationship among \(f_{\text{thing}}, f_{\text{fog}}\) and \(f_{\text{cloud}}\) is hold:

\[
f_{\text{thing}} \leq f_{\text{fog}} \leq f_{\text{cloud}}\]

(15)

where \(f_{\text{thing}}, f_{\text{fog}}\) and \(f_{\text{cloud}}\) are represented the processing speed respectively. Among all types of tasks, the processing latency of computation tasks can be varied due to they can be processed at either fog tier or cloud tier. For better describe such delay, we employ M/M/n queueing model [35] to describe the computation resources at both fog and cloud tiers. In this model, the average computation delay (waiting time plus service time) \(D_{\text{comp}}\) for processing a unit of data is described as below:

\[
D_{\text{comp}} = \frac{1}{n\mu - \lambda}.
\]  

(16)
where $n$ is the number of processing units, $\lambda$ and $\mu$ are the traffic arrival and service rate respectively. We can simply convert $\mu$ to $f$ by using the below equation

$$\mu = \frac{f}{K} \quad (17)$$

where $K$ is in terms of (# of cycles)/task.

Similar to the processing latency, most of the transmission latency in our CoT is predictable and only related to the communication overhead of the edge and the capacities of the used communication link. However, network delay and packet losses are two critical issues in non-deterministic packet networks, such as the Internet used to transmit data in between the fog tier and the cloud tier. Modelling the Internet delay and losses in an accuracy way is not trivial, for the simplicity of the analysis, we adopt the Internet delay models used in [35]. Let $d_{ij}$ denote the average delay of the WAN transmission path from the fog device $i$ to the cloud server $j$. Thus, when the traffic rate dispatched from the fog device $i$ to the cloud server $j$ is $\lambda_{ij}$, the corresponding communication delay $D_{ij}^{comm}$ is:

$$D_{ij}^{comm} = \lambda_{ij}d_{ij}. \quad (18)$$

5. Simulation and proof-of-concept

In this section, a performance evaluation of our three-tier CoT system is presented, aiming at assessing the energy consumption of the whole CoT system and the service latency of applications. Moreover, a comparative study of both two-tier and three-tier CoT systems is provided, also in terms of the energy consumption of the whole CoT system and the completion time of applications.

Finally, in order to demonstrate the feasibility and illustrate the functioning of our models in practice, we implemented a Proof of Concept (PoC) in real nodes.

5.1. Simulation setup

In order to evaluate the performance of both two-tier and three-tier CoT systems, we carried out a discrete event simulation (DES) making use of the SimPy [14]. SimPy is a process-based discrete-event simulation framework based on standard Python programming language (version 2.7.11) [28].

Our simulation was designed as a numerical study that utilizes the high-level events to collect the results rather than aiming to implement all the lower level network details. This is due to the main focus of this paper is to evaluate how the CoT system performance is affected by the involvement of fog computing. A desktop computer equipped with an Intel Core 2 Duo 2.80 GHz processor and 4 GB RAM was used to run all the simulations in a controlled environment within the wireless networks laboratory (LabNet), at the Federal University of Rio de Janeiro. Each experiment was performed under 30 repetitions, what provided a reasonable confidence interval of 95% for the obtained results.

As mentioned before, each application may include four types of tasks, namely sensing tasks, computation tasks, communication tasks and actuation tasks. Please note that, the sensing and actuation tasks in our simulation can be only performed at the things tier, while computation tasks can be performed at fog or cloud tier, and communication tasks can be performed at all tiers. By varying the computation tasks allocation strategy (e.g. varying the percentage of computation tasks allocated at fog or cloud tier) of the same set of applications, we can observe how the system performance changes accordingly. Before introducing further
For deploying PEs in our environment, considering the total sensing area \( G \), we used the following procedure. We initialized physical topologies by randomly (uniformly) choosing the position of the single access point of each area \( O_i \) in \( G \), considering a Cartesian plan. The access point coverage radius defines a circle, centred at the access point, which defines the respective area, and within which are all the PEs pertaining to the given area. Then we choose random polar coordinates \((r, \theta)\) for each new PE added to this area, where the \( r \) coordinate is chosen randomly and lower than the area coverage radius (uniform probability), while \( \theta \) is randomly chosen from \([0, 2\pi]\) (also uniform probability). Then we converted the polar coordinates of the recently added PE to Cartesian coordinates, and got the final positions of PEs, in terms of the latitude and longitude of PEs in \( G \). We considered that each access point of an area directly connects to a fog node in its position. Therefore, the numbers of access points, areas and fog nodes are the same.

Regarding the IoT applications, we modelled them as “chains of tasks”. Each given application contains one or more chains. Each chain contains a sequence of tasks in which each task has at most a single predecessor and successor task. Sensing, computation and actuation tasks are sequentially connected by communication tasks.

For generating applications, the following procedure was employed. We initialized an application by determining the total number of task chains it contains. The first task of each chain has to be a sensing task. The sensing tasks of each application are always processed at the things tier. After the sensing tasks, an amount of computation tasks (given by the “number of processing tasks per chain” in Table 1) are added in each chain with the respective communication tasks in between them, and these tasks could be performed at different tiers. Next, after the computation and communication tasks were created, an actuation task is added to the end of each chain. After all the tasks were created, the workload of sensing, computation and actuation tasks and the overhead of communication tasks were randomly generated according to the exponential distributions with the respective lambda values shown in Table 1. In our simulations, we were particularly interested in applications whose communication overhead is greater than the computation workload.

In our simulations, for the ease of the analysis, we considered the following behaviour of applications and their tasks. When the application arrived in the cloud, the sensing tasks of it would be distributed immediately. These tasks are always at the beginning of a chain, and must be transmitted to the respective PE that will perform it. The computation, communication and actuation tasks of the same chain were distributed as a consequence of the completion of the first sensing task of the chain. Moreover, the whole chain is performed only once during the simulation duration. After the completion of all the chains of an application, the application ends, and is removed from the system. As mentioned before, the computation tasks of each chain can be allocated to different tiers. For these tasks, we set up a parameter called percentage of computation tasks offloaded (PTOF). The value of PTOF represents the percentage of the computation tasks in the chain that is allocated to the fog tier. For each computation task, its location in the chain is determined by using its own

Fig. 6. Results of (a) S-MR, (b) S-AD, (c) H-MR and (d) H-AD metrics with different speed ratio.
weight compared to the overall weight of the computation tasks in the chain. Once the value of PTOf is determined, a separator is generated and the computation tasks are divided into two parts according to their execution sequences in the chain. The tasks that are located before the separator will be performed at the fog tier. The remaining computation tasks of the chain will be performed at the cloud tier. Moreover, our decision mechanism in the simulation assigns sensing tasks to virtual entities randomly and uniformly, so that every virtual entity has equal probability of being chosen to perform a task. In our simulations, the arrival time of the applications is randomly (uniform) distributed in between the start time of simulation (simulation time = 0) and the total simulation time (duration).

5.2. Evaluation methodology

Considering the objectives of the performed simulations and the Goal Question Metric (GQM) methodology introduced in [30], the goal (G1) of our studies is to assess how the integration of fog computing with the traditional two-tier CoT affects the system performance. Such goal is further refined in two questions, Q1 and Q2.

Q1: Does the involvement of the fog tier reduce the completion time of applications?

Q2: Does the involvement of the fog tier affect the energy consumption of the CoT system as a whole?

The performance metrics used to answer the above questions are summarized in Table 2.

Regarding Q1, we employed four different metrics, including soft deadline missing ratio (S-MR), average distance to soft deadline (S-AD), hard deadline missing ratio (H-MR) and average distance to hard deadline (H-AD). In here, the soft deadline means the deadlines associated with the tasks can be occasionally missed. The result of missing such deadline can be tolerated and the application still remains operational. For the hard deadline, we mean that meeting the deadline is critical for the application functionality and missing it will lead to undesirable or even catastrophic consequences. The S-MR is defined as the ratio of the number of tasks missing soft deadlines during the application execution in relation to the total number of soft deadlines needed to be met. In other words, the S-MR can also be recognized as the ratio of the number of late tasks (tasks which had their soft deadline missed after running in simulation) in relation to the total amount of task existing in the simulation. A soft deadline is considered as missed when the actual completion time of a task is greater than its assigned deadline, which is the expected completion time of that task. Since the outcome of the execution of late tasks may be useless to the end users, the S-MR is a metric that reveals objectively how useful, from the tasks viewpoint, is our proposal regarding the assignment of tasks to the fog tier. In turn, the S-AD metric reveals how our task allocation scheme (the proportion of computation tasks assigned to the fog tier) can allow more tasks to meet their deadlines. For defining S-AD metric, it is first necessary to define the “distance to the soft deadline of a task” as follows. For a given task \( v \), let \( S_v \) be the value of the soft deadline assigned and let \( F_v \) be the actual completion time of the same task \( v \). The distance to the soft deadline of a given task \( v \) is calculated by \( D_v = S_v - F_v \). The S-AD is then denoted as the average of the values of \( D_v \) calculated for all tasks and is measured in units of time. The H-MR and H-AD metrics have definitions similar to S-MR and S-AD and they are used to measure the hard deadlines of the tasks. Although some soft deadlines of tasks of a given application may be missed during the execution, the application still has chance...
to be completed within the required deadline if all tasks can be completed before their hard deadlines. The H-MR is defined as the ratio of the number of tasks that missed hard deadlines during the application execution in relation to the total number of hard deadlines existing in the simulation. For the metric of H-AD, we define the “distance to the hard deadline of an application” as follows. For a given application $a$, let $S_a$ be the value of the hard deadline assigned and let $F_a$ be the actual completion time of the application. The distance to the hard deadline of a given application is calculated by $D_a = (S_a - F_a)$. The H-AD is then denoted as the average of the values of $D_a$ calculated for all tasks and is measured in units of time.

Regarding Q2, we used the following metrics: total energy consumption by the things tier (TESl), total energy consumption by the cloud tier (TESc), total energy consumption by the fog tier (TESf), total energy consumption by the whole system (TESs), average energy consumption by a PE in the things tier (AESf), average energy consumption by a node in the fog tier (AESf), maximum energy consumption by a PE (MESt), and the lifetime of a PE (LPE).

The TESl is defined as the sum of the energy consumption of the PEs during the entire simulation. We calculated the sum of the energy consumption for each PE in terms of transmission and reception of messages (Eqs. (6), (7)); sensing (Eq. (5)); actuation; and processing (Eqs. (8), (9)). TESf is the sum of the energy consumption of the fog nodes during simulation, the static and dynamic energy consumption (Eqs. (11), (12)) are considered. TESc is calculated similarly, but for the cloud tier, both the static and dynamic energy consumption and the energy consumption for cooling (Eq. (14)) are considered. TESs is the sum of TESl, TESf and TESc. AESl is calculated as the TESl divided by the total number of PEs in the things tier. Similarly, AESf is calculated as the TESf divided by the total number of nodes in the fog tier. The MESt is defined as the maximum energy consumption of a specific PE among all PEs during the simulation. In comparison to AESl, which reveals how nodes spend energy in average, MESt reveals the PE that spends the most energy in the things tier. Considering that a device instrumenting a PE is normally powered by batteries, it is possible to infer this particular PE will be the first one to have its battery depleted. For this reason, we define LPE, assuming that a common energy source of a PE is two AA batteries, capable of providing 16 kJ during the lifetime of the PE. The lifetime of a PE can thus be calculated as $LPE = 16 \text{kJ}/(\text{MESt}/\text{simulation time})$.

5.3. Experimental results

This sub-section provides the detailed results of our experiments, in which the odd-numbered experiments relate to question Q1 and the even-numbered experiments relate to Q2. All designed experiments are divided into four sets for measuring how the system performance is affected by different factors, namely, the number of applications, the speed ratio of cloud to fog, the transmission delay and the packet loss rate. For all these experiments, we investigate the variation of system performance under different PTOF (the percentage of computation tasks is performed at fog tier) rates, which is ranged from 0% to 100%. In addition, we use PTOF = 0% to represent the performance of any two-tier CoT system, so this is considered a benchmark for our results. This is due to no computation tasks are allocated to the fog tier so that such a three-tier system can be considered as an analytic approximation of a two-tier system, even it has more access points than the actual two-tier CoT systems. It is important to mention that only computation tasks can be allocated to different tiers, and no sensing or actuation tasks can be allocated to the fog or cloud tier.

---

5.4. Experiment E1

The first set of experiments is used to study how the system performance is affected by the variation of the number of applications. The results of E1 related to Q1 are shown in Fig. 3. Please note that, since the legend of all figures is the same, we put it outside for better displaying and this setting will apply to all of our experimental results.

As shown in both Fig. 3(a) and (c), the values of S-MR and H-MR are inversely proportional to the PTOF rates. When all computation tasks are performed in the cloud tier (PTOF = 0%), both S-MR and H-MR reach their peak. When all computation tasks are performed in the fog tier (PTOF = 100%), the S-MR and H-MR have their lowest values. Along with the number of applications increasing, the results have no significant changes for all PTOF rates due to the infinite resource setting of the cloud tier and the high-density deployment of fog devices in the simulation. In Fig. 3(b) and (d), we can observe that the values of S-AD and H-AD are directly proportional to the PTOF rates. These results indicate that more tasks from the same set of applications can be finished before the assigned soft deadlines in the three-tier CoT system than in the two-tier CoT. There are multiple factors that contributed to this outcome. The first factor relates to the fact the computation tasks of the applications can be processed by the nearby fog nodes instead of being transmitted to the far-end cloud servers, especially for those applications that are associated with heavy communication overhead tasks. The second one is that the fog nodes only provide their services to the PEs located at the same area, but the cloud needs to handle the service requests from the clients located at different areas. With the large amount of incoming requests, the resource on the cloud tier can be consumed very quickly and some tasks may not be immediate served from the cloud.

5.5. Experiment E2

The results of experiment E2 related to Q2 are shown in Fig. 4 (results of TESf, TESc and TESs) and Fig. 5 (results of AESf, AESc, AES and LPE) respectively.

Fig. 4(a) shows that the values of TESf are proportional to the number of applications. However, the nearly identical trend of TESf for all PTOF rates suggests that any change in the computation tasks allocation decision onto the fog tier does not obviously influence the energy consumption in the things tier. Regardless how the percentage of PTOF changes, the energy spent for sensing, processing (data acquisition and aggregation) and actuation will remain the same, which is mainly determined by the number of sensing and actuation tasks required by each application. Also, the energy consumption for communication will be the similar case, which is determined by the needs of transmission to the access points (after the end of execution of sensing tasks) and by the needs of transmission from the access points (before the execution of actuation tasks).

Fig. 4(b) shows that the values of TESf are proportional to the number of applications, as well as the PTOF rates. When more computation tasks are allocated to the fog tier, the energy consumption of the fog tier will thus be increased since more devices are used to process the tasks. Fig. 4(c) presents the opposite energy consumption trend compared to that shown in Fig. 4(b). This result is obvious since more cloud servers are used to process the tasks if fewer tasks are allocated to the fog tier. However, the
energy consumption of the cloud tier is proportional to the number of applications, which is similar to the trend shown in Fig. 4(b).

Fig. 4(d) indicates how the overall energy consumption of the system is affected accordingly when the number of applications increases. First of all, the trend of TESs is opposite to the PTOF rate. However, the trend of TESS is similar to the trend of the TESC since the energy consumption of the cloud tier is the major part of the CoT systems. More importantly, we found that when more computation tasks are allocated to the fog tier, the amount of energy saved in the cloud tier is greater than the extra energy spent in the fog tier. Thus, the three-tier CoT model can reduce the energy consumption of the entire system compared to the traditional two-tier CoT model. Finally, the values of TESF and TESC are $10^6$ times bigger than the values of TESl, so that the TES has little influence to the TESs. Despite of this, the PEs have limited power sources in contrast to other tiers. Thus, the energy spent by each PE directly influences its lifetime, as well as the system lifetime. In the following, we show results in Fig. 5 which are related to an assessment about the lifetime of PEs.

Fig. 5(a) shows that the AESl is proportional to the number of applications. However, when the number of applications is fixed, it has almost the same value no matter how the PTOF rate changes. Fig. 5(b) shows that the values of AESF grow almost linearly when the number of applications increases. Also, the AES is proportional to the PTOF rate. The higher PTOF rate leads to the larger AES.

Fig. 5(c) shows that the trend of MESS is proportional to the change of the number of applications. However, the change of MESS is not significantly affected by the PTOF rate. By observing the change of MESS, it is possible to conclude that no significant effect is found on the maximum energy consumption of a single node at the things tier with the involvement of the fog tier. In comparison to the results of AESl, we observe that the value of AESl is stable when the PTOF rate is changed and the energy consumption is balanced among the nodes in things tier. This is a natural result of the mechanism used for assigning sensing tasks to nodes at the things tier, in which the tasks are randomly assigned to the nodes with an uniform probability. The values of MESS are then used in the later evaluation of the LPE metric. Fig. 5(d) shows that LPE is inversely proportional to the change of the number of applications, which indicates that the more applications are run on the system simultaneously, the less lifetime of the things has. Similar to the result of MESS, the significant performance difference among different PTOF rates is not observed. Therefore, the percentage of computation tasks allocated to the fog tier has no significant contribution to the estimated lifetime of nodes at the things tier since we assume that all the computation tasks are always performed at either fog tier or cloud tier.

5.6. Experiment E3

This set of experiments aims to study how the system performance is affected by the variation of speed ratio of cloud to fog. The results of E3 related to Q1 are shown in Fig. 6.

As shown in Fig. 6(a) and (c), the values of S-MR and H-MR for PTOF = 0% are affected by the change of the speed ratio of cloud to fog. However, the changes are not very significant in our simulation since the applications could be comprised of heavy communication tasks and light computation tasks so that the communication tasks consume most of the execution time within the deadline, and the computation tasks may not be able to be processed on the cloud immediately. These results suggest that in order to reduce the completion time of the applications, we need to not just improve the processing speed of the servers, but also to ensure the tasks can be processed as soon as possible (upon their arrival). In Fig. 6(b)
and (d), the S-AD and H-AD demonstrate this result from a different aspect.

5.7. Experiment E4

The results of experiment E4 related to Q2 are shown in Figs. 7 and 8, respectively.

Fig. 7(a) shows that energy consumption of the things tier do not change visibly when the speed ratio is increased. For any given PTOf rate, the variation of TESf is within 10 J at all times. As shown in Fig. 7(b), the value of TESf does not change significantly when the speed ratio is increased. However the TESf is still directly proportional to the PTOf rate, the higher assigned PTOf ratio thus consumes more energy in the fog tier. Fig. 7(c) indicates how the energy consumption is varied accordingly to the change of speed ratio. As clearly shown in the figure, the energy consumption of cloud is significantly affected by the speed ratio. This is due to when the speed ratio increases, the same amount of computation workload can be completed within a shorter period of time. Consequently, the energy consumption is reduced not just because the servers can be turned into the sleeping mode for energy saving once they have no assigned ongoing tasks, but also the related cooling infrastructure can be temporarily turned off if the related servers are put into the sleeping mode. As indicated in the previous experimental results, the main energy consumer of the system is cloud. Therefore, the energy consumption for the entire system appears to show the similar trend as the cloud, which is given in Fig. 7(d).

Fig. 8(a)–(c) show that the energy consumption of the PEs in the things tier is not significantly affected by the speed ratio and the average energy consumption of the fog tier is directly proportional to the PTOf rate. Lastly, the PE lifetime shown in Fig. 8(d) is not very significantly affected by the change of speed ratio since the lifetime is mainly determined by the energy consumption of the things tier, not the processing speed provided in either the fog tier or the cloud tier.

5.8. Experiment E5

This set of experiments is used to investigate how the system performance varies when the transmission delay is changed within the system. The results related to Q1 are shown in Fig. 9.

As shown in Fig. 9(a) and (c), the values of S-MR and H-MR are greatly affected by the change of the transmission delay. When all computation tasks are performed in the cloud tier (PTOf = 0%), both S-MR and H-MR have the highest values and produce the most marked increments among all PTOf rates when the transmission delay is increasing. Compared to it, the three-tier CoT with different PTOf rates seems to be less affected by the transmission delay than the two-tier CoT, and the S-MR and H-MR of the three-tier CoT are inversely proportional to the PTOf rates. In Fig. 9(b) and (d), the results of S-AD and H-AD fully support the above discussion, since the average application completion time is increased as the transmission delay becomes larger and larger in the system. It is worth to mention that when the transmission delay reaches 1.9, the two-tier CoT appears to be not able to complete the applications within the average deadlines since the value of H-AD becomes negative.

5.9. Experiment E6

The results of experiment E6 related to Q2 are shown in Figs. 10 and 11.
As shown in Fig. 10(a), the energy consumption of the things tier is not significantly affected by the increasing transmission delay since the energy consumption of data transmission is determined by the size of data and the transmission distance, as represented in Eq. (6). Besides, the energy consumption of both fog tier and cloud tier is not significantly affected by the transmission delay either. As expected, the energy consumption of the fog tier is directly proportional to the PTOF rate, but the energy consumption of the cloud tier is inversely proportional to the PTOF rate. According to what we have discussed before, the energy consumption of the entire system is mainly determined by the cloud tier, which is shown in Fig. 10(d).

Fig. 11(a) and (b) present the similar trends as we have discussed in Fig. 10, where the energy consumption of the things is not significantly affected by the transmission delay and the average energy consumption of the fog tier is directly proportional to the PTOF rate. As indicated in Fig. 11(c), the maximum energy consumption of a single node in the things tier is not sensitive to the change of transmission delay. Lastly, the PE lifetime is not significantly affected by the transmission delay since the lifetime is mainly determined by the energy consumption of the things tier.

5.10. Experiment E7

This set of experiments is used to study how the system performance varies when the quality of the communication channel is changed. The results of E7 related to Q1 are shown in Fig. 12.

As shown in Fig. 12(a) and (c), the values of S-MR and H-MR are greatly affected by the change of the transmission delay. When all computation tasks are performed in the cloud tier (PTOF = 0%), both S-MR and H-MR have the highest values and produce the most marked increments among all PTOF rates when the transmission delay is increasing. Compared to the results shown in Fig. 9, the negative impact of packet loss is more significant than the transmission delay since the data has to be retransmitted from the sources to the destination to ensure data accuracy and freshness in our simulation. In Fig. 12(b) and (d), the results of S-AD and H-AD provide extra support for the above observation, since the average application completion time is quickly increased as the transmission delay becomes larger and larger in the system. The poor communication channel made hard for the two-tier CoT to complete the applications on time. With the increment of packet loss rate, around 60% of applications could not be completed on time within the two-tier CoT, while the three-tier CoT was still able to assure a reasonable application completion rate for the same set of applications.

5.11. Experiment E8

The results of experiment E8 related to Q2 are shown in Figs. 13 and 14, respectively.

As shown in Fig. 13(a), the energy consumption on the things tier is significantly affected by the packet loss rate, since the increasing probability of data retransmission causes the extra energy consumption at this tier. However, the energy consumption of both fog and cloud tiers is not clearly affected by the packet loss rate and the energy consumption for the fog and cloud tiers are still inversely/directly proportional to the PTOF rate. Therefore, the energy consumption of the entire system follows the energy consumption trend of the cloud tier.
Fig. 13. Results of (a) TES, (b) TESf, (c) TESc and (d) TESs metrics with different packet loss rates.

As shown in Fig. 14, the average energy consumption of the nodes in the fog tier is not affected by the packet loss rate. However, the average energy consumption and the maximum energy spent in the things tier are directly proportional to the increasing packet loss rate. As a result, the PE lifetime is reduced when the packet loss rate is increased since the energy consumption of PEs is increased.

5.12. Discussion of simulations

Responding to Q1, the involvement of the fog layer effectively reduces the completion time of applications at most times compared to the two-tier systems. By changing the PTOF rate from 0% to 100%, the four designed performance metrics (S-MR, S-AD, H-MR and H-AD) of the proposed system have improved accordingly. However, these improvements were not always significantly shown in all our experiments, especially when the non-communication factors (e.g. the speed ratio of cloud to fog) were being tested. Therefore, we believe the communication-related factors can effectively improve the overall system performance in three-tier CoTs, so as to obtain more benefits in terms of a reduction in the completion time of applications.

Responding to Q2, the energy consumption of the different tiers is affected in different ways within our proposed three-tier CoT system. Firstly, the things tier is not significantly affected by the PTOF rates. This is a consequence of our simulation setting, where we deliberately created the best scenario for the two-tier system by adopting the same number of access points as the three-tier system. Besides, we also assume the computation tasks are always performed at the fog tier or the cloud tier, which makes the things tier to be only responsible for the data collection and transmission. Secondly, the energy consumption of the cloud tier is inversely proportional to the PTOF ratio since it spends less energy on task processing when more computation tasks are allocated to the fog tier. As a result, the energy consumption on the fog tier is directly proportional to PTOF rate. However, for the energy consumption of the three-tier system, this negative impact over the fog tier is beneficial, since it is lower than the effect of energy saving in the cloud tier.

5.13. Proof-of-concept of the three-tier system model

In order to demonstrate the feasibility of executing the proposed models on real sensor devices and to illustrate their functioning in practice, we built a Proof of Concept (PoC) implementation. The main goal of the implementation was to show that for time sensitive applications, the processing of the sensor generated data in a fog node, located closer to the user and/or the phenomenon being monitored, instead of in a remote cloud sensor, is much more effective. We implemented a simple scenario, with a setup similar to the one used in the simulations.

For the things tier, we implemented a wireless sensor network (WSN) composed of nine nodes. These WSN nodes represented the Physical Entities (PE) in our PoC. The WSN platform used was Zolertia Z1 [44]. Each Z1 node is composed of a single-chip 2.4 GHz IEEE 802.15.4 compliant radio transceiver, called CC2420, a low-speed processor (20 kHz), model MSP430, a thermometer from Texas Instruments, model TMP102 (representing the sensing device of the PEs), and two AA batteries as energy source [45]. Eight Z1 nodes played the role of instrumenting devices, sensing temperature values at the target area. One if the nodes played the role of the WSN sink node or base station, representing the interface between the WSN and external systems/networks. The sink node was plugged to a PC, acting as the access point to
communicate the things tier to the cloud or fog tier. The logical topology of the WSN is shown in Fig. 15.

For the fog tier, we used a single PC connected to the intranet of our laboratory, with the following configurations: Intel i5-2500k processor, quad-core 3.3 GHz and 8 GB of RAM, running a Debian 7 as operational system.

For the cloud tier, we used a server with the same configurations of the fog tier, but running an instance of the software platform for cloud computing OpenStack [46]. This server is located at Centre for Distributed and High Performance Computing, The University of Sydney.

For the IoT application, we implemented a simple fire detection application in the WSN nodes. Z1 nodes use Contiki WSN operating system, version 2.7 [47], and the application was implemented in C programming language. In order to comply with our proposed model, the application was modelled as a “chain of tasks”, being composed of three chains, each one with a sensing task, a processing task and an actuation task. The complete source is available at our website (http://ubicomp.nce.ufrj.br/3tieriot/). The goal of each sensing task is to sense temperature at the position of the respective sensor node and transmit the sensed values. Each processing task (which can be executed in either fog or cloud tiers) checks if the transmitted temperature value is above a threshold (set as 50 °C). Whenever this threshold is surpassed, the actuation task is performed, i.e. an alarm is triggered in the same node that generated the value. Since we do not have actuator devices available at our lab, we emulated the alarm by turning on a green LED in the device.
We repeated the experiment 30 times and calculated the average communication delays between the fog node and the cloud tier (CD1 = 343 ms) and the fog and a node from things tier (CD2 = 283 ms). Moreover, we calculated the processing delay of sensing tasks in things tier (PD1 = 3 ms), processing tasks in fog tier (PD2 < 1 ms) and in cloud tier (PD3 < 1 ms), and actuation tasks in things tier (PD4 < 1 ms).

We consider that the entry point of the application is the cloud tier. A message must be transmitted from the cloud to the fog node (CD1), and then to the respective node in the things tier (CD2). Then the sensing task is performed (PD1), and the value is returned to the fog (CD2). In the fog node, the processing task can be performed (PD2), or the data can be transmitted to the cloud node to be processed (CD1 + PD3). In the first case, a message is transmitted to the respective thing by the fog node to perform the actuation (CD2 + PD4). In the second case, the message is transmitted to the respective thing by the cloud node to perform the actuation (CD1 + CD2 + PD4).

Thus, we have two cases. In the first case, where the processing tasks are performed in the fog node, the total delay is calculated as CD1 + CD2 + PD1 + CD2 + PD2 + CD2 + PD4. In the second case, where the processing tasks are performed in the cloud node, the total delay is calculated as CD1 + CD2 + PD1 + CD2 + CD1 + PD3 + CD1 + CD2 + PD4. The total delay for the first case is 1198 ms. The total delay for the second case is 1884 ms.

6. Conclusion

This work focuses on modelling and analysing the involvement of fog computing in the design of a CoT (Cloud of Things) infrastructure. The goal of this paper is to develop a mathematical model of three-tier CoT to assess the applicability of the fog tier in the context of the system and to demonstrate it is a key factor to meet the demands of the time-constraint applications. Results clearly depict the enhanced performance of the CoT system architecture including the fog tier both in terms of meeting applications deadline and saving energy under such situations.

In the future, we first plan to extend this work by proposing a multi-objective task allocation scheme to support real-time implementation as well as some other QoS (Quality of Service) requirements. Moreover, we plan to investigate some other aspects of the regional fog presented in our model. A different number of access points will be added to the regional fog. We will also employ more complex and realistic communication models in between PEs and access points, so that the data collected by each PE can be sent to rear tiers by choosing the best route. Finally, we will study whether regional fog can be beneficial from using more localized communication for reducing mean service latency of applications and even reducing the energy consumption at the things tier.

Acknowledgements

Dr. Wei Li’s work is supported by Faculty of Engineering and IT Early Career Researcher scheme, The University of Sydney, and the Faculty of Engineering & Information Technologies, The University of Sydney, under the Faculty Research Cluster Program. Igor L. Santos’s work is supported by a scholarship from CAPES Foundation, Ministry of Education of Brazil. This work was partially supported by Brazilian Funding Agencies FAPERJ (under Grant E_06/2015E_06/2015 for Flavio C. Delicato) and CNPq (under Grants 200757/2015-6 and 307378/2014-4 for Flavio C. Delicato, 310958/2015-6, 457783/2014, and 200758/2015-2 for Paulo F. Pires). This work is partly supported by the Brazilian Funding Agency CNPq under Grant Number 304527/2015-7. This work is supported by West Virginia Higher Education Policy Commission under Grant FR12W752W. This work is supported by NSFC Grant (Program Nos. 11301414, 61472318 and 11226173). Prof. Zomaya’s research was supported by funding from the Faculty of Engineering & Information Technologies, The University of Sydney, under the Faculty Research Cluster Program.

References

Albert Zomaya is the Chair Professor of High Performance Computing & Networking in the School of Information Technologies, University of Sydney, and he also serves as the Director of the Centre for Distributed and High Performance Computing. Professor Zomaya published more than 600 scientific papers and articles and is author, co-author or editor of more than 20 books. He is the Founding Editor in Chief of the IEEE Transactions on Sustainable Computing and serves as an associate editor for more than 20 leading journals. Professor Zomaya served as an Editor in Chief for the IEEE Transactions on Computers (2011–2014). Professor Zomaya is the recipient of the IEEE Technical Committee on Parallel Processing Outstanding Service Award (2011), the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing (2011), and the IEEE Computer Society Technical Achievement Award (2014). He is a Chartered Engineer, a Distinguished Engineer of the ACM, a Fellow of AAAS, IEEE, IET. Professor Zomaya’s research interests are in the areas of parallel and distributed computing, complex systems and green computing.

Samee Khan received a B.S. degree in 1999 from Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan, and a Ph.D. in 2007 from the University of Texas, Arlington, TX, USA. Currently, he is Associate Professor of Electrical and Computer Engineering at the North Dakota State University, Fargo, ND, USA. Prof. Khan’s research interests include optimization, robustness, and security of systems. His work has appeared in over 300 publications. He is on the editorial boards of leading journals, such as IEEE Access, IEEE Cloud Computing, IEEE Communications Surveys and Tutorials, and IEEE IT Pro. He is an ACM Distinguished Speaker, an IEEE Distinguished Lecturer, a Fellow of the Institution of Engineering and Technology (IET, formerly IEE), and a Fellow of the British Computer Society (BCS).