The international workshop on networking algorithms and technologies for IoT (NAT-IoT) 2015

Ad Hoc Cloud as a Service: A protocol for setting up an Ad hoc Cloud over MANETs

Bilel Zaghdoudia, Hella Kaffel-Ben Ayedb,c, Imen Riabic

LIPAH, Department of Computer Science, Faculty of science of Tunis, University of El Manar, Tunis 2092, Tunisia

Cristal Laboratory, National School of Computer Science, University of Manouba, Tunisia

Department of Computer Science, Faculty of science of Tunis, University of El Manar, Tunis 2092, Tunisia

Abstract

Mobile and wireless networks permit to widen traditional internet applications to this kind of networks. Furthermore, with the dramatic increase in their capabilities, mobile devices are becoming one of the main processing devices for users. However, they still lack resources compared to traditional processing devices. Integration between mobile devices over mobile ad hoc networks and Cloud computing is addressed in various studies. The goal of this paper is to propose a protocol for the deployment of an ad hoc mobile cloud over MANETs. This scenario has multiple advantages such as ubiquity, availability, affordability, opportunity and spontaneity. The proposed system is composed of two types of nodes: Providers and Customers. A group of nodes with sufficient resources (Providers) collaborate to set up an IAAS like cloud and provides their services to a set of nodes (Customers) requesting resources. The protocol permits a dynamic management of provider and customer nodes.

1. Introduction

Using remote Cloud would permit to overcome the lack of local resources. However, this type of service and resource providing depends on the connection to the remote cloud, and the system may fail in low connectivity scenarios. The main solution proposed is to use the surrounding mobile devices (mobile phones, PDAs, laptops etc.) as local resource Providers and to exploit their capabilities as a mobile cloud in a different view than traditional ones.

In this paper, we present a protocol for deploying in an ad hoc manner a mobile cloud composed of two types of nodes: provider and customer. This protocol governs the interaction and the communication between Ad Hoc nodes and provides the dynamic management of providers and customers. The remainder of the paper is organized as follows; Section 2 presents the literature review related to the addressed topic. Section 3 depicts the proposed architecture and protocol. Section 4 describes briefly the design of the protocol. Finally, we conclude and present our future work in section 5.

* Corresponding author. Tel.: +216 99 916 020.
E-mail address: zaghdoudibilel@gmail.com
2. Related work

The mobile cloud computing (MCC) concept has been addressed recently by various researchers and different definitions and approaches of MCC have been proposed. The first and the most common approach of mobile cloud is to run applications on a remote resourceful server, so the mobile device acts only like a thin client connected to the remote server through 3G. Cloudlet is the second concept of mobile cloud computing, where the mobile device offloads its workload to a local Cloudlet comprised of a more powerful computer with connectivity to the remote Cloud servers. These Cloudlets would be situated in common areas such as coffee shops so that mobile devices can connect and function as a thin client to the cloudlet as opposed to a remote cloud server which would present latency and bandwidth issues. The third approach is to consider and use interconnected mobile devices in the local vicinity as resource providers aiming to provide resources or sensed data to neighboring nodes in needs. This model introduces the concept of ad hoc mobile cloud (ad hoc MCC) over P2P networks. Bluetooth is often presented as an underlying communication technology thanks to its advantages, mainly low radiation and low energy cost. However, because of the limited transmission range of Bluetooth, Wi-Fi is considered as a good substitute for Bluetooth. Wi-Fi has a high bandwidth, which allow a faster data transfer between mobile nodes and a relatively low network latency. Compared to 3G, mobile Cloud computing offloading using Wi-Fi consumes much less energy for different amount of shared jobs. The work presented in this paper follows the third approach.

A spontaneous ad hoc network can be defined as a special case of MANET that is formed by a set of mobile nodes in a closed location and during a certain period. All the nodes are free to join or to leave the group without any permission. There is no centralized control or coordinator and without user intervention. presented the five main challenges of spontaneous networks such as the network is not planned and its boundaries are poorly defined, hosts are not preconfigured, there are not any central servers and users are not experts. In, the authors studied spontaneous wireless ad hoc networks and conclude that all the nodes in the network are responsible of the management aspects and that cryptographic techniques is an important choice to maintain the security of such networks by discussing the advantages and the disadvantages. For mobile cloud computing research based on spontaneous Ad Hoc network, we can mention the research work undertaken in. It propose a trusted algorithm for creating spontaneous network, and a set of developed and tested algorithms for the management of nodes. To guarantee the security of the network, they developed an algorithm based on AES implementing a symmetric encryption scheme. The results of their work show that the performance of the network can be maintained.

In, the authors present a preliminary design for a framework to create Ad Hoc Cloud computing providers. The presented architecture is composed of five main components: Application manager; Resource manager; Context manager; P2P component and an Offloading manager. This architecture allows a user at a stable place to listen for nodes in the network presenting available resources to execute a task that his own device cannot perform. The system intercepts the application loading, modifies it in order to use the virtual Cloud. Authors in, present a framework that aims to determine the usefulness of sharing workload at runtime. Their framework is composed of three components: Resource handler, Job handler and Cost handler. The first component is a resource-discovering component, which establish connections and exchange Meta data with clients. The Job handler is in charge of recording tracking information about clients. The last component, which is the cost handler, is used to estimate costs and select suitable client devices. The authors based their implementation and experiments on Bluetooth and a master-slave system. Another architecture for job sharing in mobile Cloud computing is presented in. It is composed of a Cloud Client which is the master component charged of offloading all jobs to the central server, which is the second component acting as a resource manager. The third component is the rest of the Cloud, composed of interconnected devices. It is in charge of executing jobs received from the central server then submitting results to the master. Management of dynamic resource provider systems and algorithms for mobile Cloud computing is still in its infancy. All the studies presented above are preliminary and considered Bluetooth as underlying communication technology. However, they have not taken into consideration the networking aspects and its impact on mobile Cloud performance. None of them have considered the deployment of MCC over Mobile ad hoc network (MANETS). This can enhance the mobility and the flexibility of the system with a higher network bandwidth for faster data transfer between mobile nodes.
3. A proposal for a communication protocol and architecture

The work undertaken focuses on the deployment of P2P Cloud on ad hoc mobile networks. This type of cloud is promising in all locations such as stations, airports, museums, cafes where an ad hoc community can be gathered. A set of MCC applications and motivating scenarios are presented in\textsuperscript{13,14}. This form of cloud could be also profitable in a battlefield, natural disaster\textsuperscript{15,16}, and mountainous region and more generally in any hostile environment where connectivity with a Cloud server does not exist. When Wi-Fi terminals do not exist or are damaged, there is only mobile terminals communicating in peer to peer and evolving as a virtual cloud to provide information collected with their sensors and locally processed.

3.1. Description of the Cloud architecture

The architecture of the ad hoc mobile cloud, aims to exploit the resources of mobile terminals in a MANET network to create a virtual cloud to meet the growing needs in terms of mobile community resources. Indeed our Cloud system is composed by two main entities: Provider nodes acting as Cloud servers, and consumer nodes acting as Cloud Customers. The role of the provider nodes is to offer services to the Customers such as tasks execution and data storage, or collecting information. The communication between the different nodes is done using the Wi-Fi as a wireless communication technology. First, we define the following assumptions:

- Customer nodes must be connected by one or more service Providers.
- A service Provider must have at least one Cloud Customer.
- A cloud node can only be a Customer or a Provider.
- Two possible types of relationships exist in the mobile Cloud: "Customer Provider" or "Provider - Provider".
- The communication between two Cloud nodes is bidirectional.

Our mobile cloud is presented in Fig. 1. The Cloud Provider System (CPS) is composed of interconnected nodes aiming to provide resources to Customer nodes. This system is created in the Setup phase by the initiator (a Customer) and nodes from the Ad hoc network can join the CPS when they receive a request for the addition of new Providers to share their resources or to join the MCC System as a Cloud Customer.

3.2. Identification of the needs related to the functioning of Cloud

To ensure the proper functioning of our ad hoc mobile cloud, we first identify its requirements in terms of deployment, dynamicity and management. These will be considered in the design of our solution. the following requirements are identified:

- Synchronous communication mode;
- On-demand Cloud deployment;
- Group communication;
- Dynamic management of Cloud member (Provider, Customer/Join, Departure);
- Genericity against frameworks.

3.3. The proposal

We propose a protocol for the deployment and management of nodes in a mobile Ad hoc Cloud over MANETs. Indeed our protocol satisfies the MCCs needs that we have identified in the previous section. Specifically in Fig. 2, we will introduce a meta-layer of the application layer, called C-Protocol, located between the protocol stack TCP/IP and the Cloud framework. This meta-layer includes our proposed protocol and it should be introduced to each node member of the MCC, its role is to provide the framework Cloud with the required services. C-Protocol provides a set of services that allow both a better MCC management and to abstract and simplify a part of MANET networks operation. Among its main services:
• On-demand Cloud deployment: it allows each node in the ad hoc network to initiate the deployment of an MCC, to create a group of voluntary providers nodes (CPS).
• Dynamic management of Cloud service Providers: to allow the addition of new Providers to meet Customers needs in terms of resources, or the removal of unattainable ones.
• Dynamic management of Cloud Customers: C-Protocol offers the possibility to nodes not member of the ad hoc Cloud to join it in order to benefit from its services and resources. On addition, each Customer is free to leave the MCC.

4. The design of the C protocol

C-Protocol uses a set of UDP messages exchanged between nodes. In the following table, we give a description of all the defined messages.

<table>
<thead>
<tr>
<th>Table 1. C-Protocol messages and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messages</td>
</tr>
<tr>
<td>Cloud Setup</td>
</tr>
<tr>
<td>Cloud_advertisement</td>
</tr>
<tr>
<td>Cloud_proposal</td>
</tr>
<tr>
<td>Cloud_setup</td>
</tr>
<tr>
<td>Add provider(s)</td>
</tr>
<tr>
<td>NewP_request</td>
</tr>
<tr>
<td>NewP_response</td>
</tr>
<tr>
<td>NewP_notification</td>
</tr>
<tr>
<td>PL_exchange</td>
</tr>
<tr>
<td>PL_Backup_request</td>
</tr>
<tr>
<td>PL_Backup_response</td>
</tr>
<tr>
<td>Add customer(s)</td>
</tr>
<tr>
<td>Cloud_discovery</td>
</tr>
<tr>
<td>Discovery_response</td>
</tr>
<tr>
<td>JoinC_request</td>
</tr>
<tr>
<td>JoinC_refuse</td>
</tr>
<tr>
<td>JoinC_accept</td>
</tr>
<tr>
<td>JoinC_notification</td>
</tr>
<tr>
<td>Cloud Setup</td>
</tr>
<tr>
<td>Node_tracing</td>
</tr>
<tr>
<td>Node_departure</td>
</tr>
</tbody>
</table>

4.1. MCC setup

This phase aims at setting up an ad hoc MCC over a spontaneous MANET. It is started by any node needing resources (the Initiator). This node initiates the Cloud setup by announcing a request on the network. The process of setting up is described the sequence diagram shown in Fig. 3.
The Initiator generates an identifier for the CPS (Cloud-ID field) and starts the cloud deployment process by broadcasting a Cloud_advertisement message. The payload data of this message carries the framework data (the criteria of selection such as the CPU, bandwidth, and battery charging status). Only nodes implementing a cloud framework will process the message since the payload data is delivered to the upper layer framework for decision. Interested nodes (whos upper layer framework component agreed to participate to the cloud) send an unicast message to the initiator i.e. Cloud_proposal message. In order to give the chance to all mobile nodes to send a proposal, the Initiator waits for a period of time to collect the Cloud_proposal messages (t-wait). The estimation of this waiting time will be determined by simulations. After the collection of Cloud_proposal messages, the Initiator generates a unique identifier for the selected providers PI (Provider identifier). The Initiator creates primary Provider and Customers lists (CL) then sends a Cloud_Setup message in multicast to the set of Providers. This message contains the Cloud parameters (PL and CL). The CL initially contains only the initiator. We define the PL as a table containing information relative to the provider nodes, including direct neighbors of each supplier and their type.

![Fig. 3. Sequence diagram of the Setup phase](image)

4.2. Dynamic management of MCC nodes

The process of adding new Providers to the cloud presented in Fig. 4 is triggered following a request made by the framework of the Cloud. To simplify the modeling of this process we will break it down into five steps. In the first step, when receiving a request to increase the resources, the various Provider nodes are launching a process of adding new Provider nodes in their neighborhood (locally). At the end of this process, Provider nodes perform a local update to their PL to add the new members. The second step is to exchange PLs from step 1 between the different service providers in order to perform a global update of PLs. At the end of this step we need to get the same PL list containing all the new members. Noting that the newly added members are not actually put into service until the reception of Cloud parameters. In the third step, after the exchange of PL lists between the Provider nodes, each one of them must verify that it has received all the PLs by comparing the number of received PLs and the number of Providers. If a Provider detects the absence of one or more PLs, it triggers a process of recovering non-received PLs. Thus all the Provider nodes possess the same PL, which guarantees coherence all over the system. Later in the fourth step, Cloud parameters will be sent to new providers. To achieve this goal, we will assume that each provider which has added new members, should be responsible about sending parameters to them. Once the new Providers receive the PL and CL lists, they begin their mission. Finally in the fifth step, the new PL should be also transferred to the Cloud Customers to realize new suppliers added. To do this, we considered the same principle of PL sharing used in the fourth step. Consequently, each service Provider is supposed to inform its Customers located in his neighborhood about the new list PL.
4.3. Add new provider process

Each Provider node identifies its neighbor nodes which are not members of Cloud and send them a message NewP_request. When receiving this message, the interested nodes responds by sending a message NewP_response containing the characteristics of their mobile terminal. Once the response collection time is over, the framework is charged by the selection of new members. Thereafter, the service Provider generates for each new Provider a unique identifier and send it in a message NewP_notification. On the reception of this message the new Providers wait for Cloud parameters while the Provider member updates its PL and creates a copy of this list called PL_Backup needed for the recovery phase that we will detail later.

4.4. PL recovery process

The process of recovering nonreceived PLs in the phase of exchanging lists between Providers aims to maintain the coherence of the system. If any of these Providers detects that one or more colleague Provider have not sent their list, it identifies them and sends to each one of them a message PL_Backup_request to retrieve PLs that have not been received previously. The Provider responds with a message PL_Backup_Response containing the PL_backup.

5. Conclusion

In this paper, we explore the deployment of P2P mobile cloud over mobile ad hoc networks. We propose the architecture of the mobile cloud and a protocol (C-Protocol) that provides a dynamic management of the mobile cloud members (Providers, Customers). The C-Protocol provides to the upper cloud framework a set of services like an on-demand deployment and a dynamic management of Cloud members. We taken into consideration the mobility and the dynamicity aspects of mobile Ad hoc network. The proposed protocol is generic with regards to the cloud framework. Future work will be focused on performing the evaluation of the C-Protocol as well as the security issues.
References