Mobile Ad-hoc Network Management in the Cloud

Hazzaa Naif Alshareef
Department of Computer Science
University College Cork (UCC)
Cork, Ireland
haz3@cs.ucc.ie

Dan Grigoras
Department of Computer Science
University College Cork (UCC)
Cork, Ireland
d.grigoras@cs.ucc.ie

***Abstract* — One benefit of creating a mobile ad-hoc network (MANET) is that members with no direct access to the Internet can avail themselves of peers' connections to the Internet and use public cloud services. Guaranteed access to cloud services is important in the case of emergency or disaster recovery situations. As managing a MANET is a difficult task, this paper introduces a new model for MANET management that uses cloud services. While preserving the ad-hoc nature of a mobile network, its management by the cloud provides reliability and robustness. The design and implementation of the system are presented, as well as a method for allowing ad-hoc network creation on Android devices. To evaluate the feasibility of the proposed prototype, an Android application was designed and tested. It offers communication with the cloud and supports file sharing among peers.**

Keywords — Mobile Ad-hoc Networks, Android Development, Cloud Computing, Mobile Cloud Computing.

I. INTRODUCTION

The ever-increasing popularity of smart mobile devices can be explained by their capabilities to access web services, provide entertainment and communicate with each other. In addition to cellular communication, smart mobile devices can use their Wi-Fi or Bluetooth radio networking cards to create or join mobile ad-hoc networks (MANETs). MANETs facilitate many applications such as file sharing and multi-player games and support less costly communication among members. For example, if one member has no direct access to the Internet, it can avail itself of the MANET to access it through other members of the network that have an Internet connection. In this project, we are interested in access to public cloud services for the members of a MANET, an aspect that might be very important in the case of emergency or disaster recovery situations. Indeed, access to site maps, including utilities, health care services, and so on, all provided as cloud services, are very important immediately after earthquakes, storms, floods or fire. One or more network member with Internet access can download and share maps, guidelines, etc. or establish video links with other members. When more than one MANET member has Internet access, the cloud can monitor the quality of the links and choose the best (in terms of signal strength and stability) for critical communication. The MANET-cloud model can have other interesting applications as well, for example in reducing communication overcrowding in dense areas (in terms of devices), and in tourism and commerce.

MANET management is a complex task, generally carried out by a middleware system running on all mobile devices and, therefore, consuming their resources [1], [2]. Specifically, MANET events such as when a new device joins, the sudden departure of a device, a network split or the merging of networks are difficult to deal with. Indeed, the process of IP allocation in order to avoid duplicates or waste consumes a lot of mobiles' resources, especially energy, as it generally involves all the MANET members. Network split due to the mobility of devices can lead to session termination between peers that now belong to departing sub-networks. The MANET management by cloud services proposed in this paper provides a reliable and robust solution to these problems.

In our model, the MANET preserves its ad-hoc nature but we assume that one or more of its members has Internet access; therefore, they can all use cloud services. In our scenario, as soon as a mobile device creates a MANET, that device will register it with a cloud service for MANET management. This service will learn about the new network, then about the members that join subsequently and the quality of communication links between the cloud and MANET members when they exist. The main purpose of this new cloud service is to monitor the MANET connection(s) to the cloud and its membership in order to maintain communication with the network in the context of its members' mobility. For example, if a mobile device providing communication with the cloud leaves the MANET, the management service can select another communication link if one is available. If there are several links, the one with the strongest signal can be chosen. The cloud management service will deal with departure, split and merge as it can use cloud persistent storage and allocate computing resources to the MANET as required during its lifetime. The implementation of this model faces challenges which are both scientific and technological. The most challenging is tackling device mobility and the dynamic topology of the MANET. Cloud resources, especially storage, can be used to maintain and process consistent information about the MANET and its individual members. The loss of connectivity between MANETs and the cloud for a shorter or longer period of time also needs to be dealt with by saving and restoring their states.

Technologically, the feature for creating Wi-Fi ad-hoc networks is blocked in most operating systems (OS) of mobile devices. The Android OS offers what it is called a tethering service, which lets users create a hotspot and then accept neighbours to connect to this hotspot. However, this is not an ad-hoc network because it requires access point (AP) configuration, whereas ad-hoc describes an infrastructure-less network. As such, there is a need to investigate other

978-1-4799-5919-8/14 $31.00 © 2014 IEEE
DOI 10.1109/ISPDC.2014.22

techniques that allow the creation of a Wi-Fi ad-hoc network with mobile Android-based devices.

In this paper, we address the challenges of MANET management and provide reliable and robust solutions to difficult operations such as join/leave, split, merge and session abortion. We also provide an approach for enabling ad-hoc networking with Android devices.

The remainder of this paper is organized as follows. Section II considers related work. Section III introduces the MANET–cloud model. Section IV analyses the implementation of the model. Section V discusses the experimental results, and section VI presents the conclusions and future work.

II. Related Work

Here, we review similar research projects or industrial initiatives relevant to our work. However, to the best of our knowledge, there is no project proposing MANET management by cloud services.

The paper [3] presents a number of network models that are based on the idea of mobile multi-hop ad-hoc networking. The paper refers to them as MANET-born models because they are built on the lessons learned from MANET research. The authors believe that there is a lack of attention to application and middleware in the MANET field that leads to a reduction in MANET spread. In a summary of the paper, the authors expect that the presented models will be mixed together and/or combined with infrastructure-based networks to create what they call “the multi-paradigm era”.

Zhang et al. propose a secure geo-social networking system that uses a Wi-Fi-based multi-hop MANET [4]. The main goal of the paper is to present a system that allows users who are located in the same place to communicate with each other using a Wi-Fi network. They can share content (photos, videos), chat, and form relationships. The paper provides a multi-hop MANET platform for mobile devices called MoNet, where two applications are designed: Wi-Face and Wi-Market. Cloud services are mentioned in the paper but not for MANET management and no implementation is presented.

In another project [5], a framework called the Mobile-Cloud, which aims to address trust management, secure routing, and risk management issues in MANETs is introduced. The framework transforms traditional MANETs to a new service-oriented communication architecture, where each member becomes a service node that can be used as a service provider or service broker. In general, each member of such a MANET has to provide sensing data about the device itself (such as battery state, processor type), and neighbouring mobile nodes (such as address, link quality, number of hops). As a result, the cloud will have an overview of each MANET operation that can be used to provide information to mobile devices for decision making (routing a request, for example). The framework intends to handle all processing and data collection in a centralized way for the purpose of turning all communication from one-to-many to one-to-one. Clearly, the paper requires each mobile device to have an Internet connection to be able to connect to instances in the cloud. That is not possible for most MANETs.

Recently, Android was provided with a new service called *Wi-Fi Direct* [6], which allows users to create one-to-one or one-to-many communication without the need to set up an infrastructure. Wi-Fi Direct uses a peer-to-peer (P2P) framework where one device acts as group owner and any other device can be a peer to that device. It relies on a technique called Soft-AP (software enabled access point), which makes a wireless antenna work as both access point and client [7].

Wi-Fi Direct is offered in most new devices such as mobile phones, cameras, printers, PCs, and gaming devices. According to one project [8], there were more than 1,400 different devices certified for Wi-Fi Direct as of October 2012.

Android developers provide a Wi-Fi Direct feature in Wi-Fi-enabled devices: Android 4.0 (API level 14) or later. Programmatically, developers can use P2P and a Wi-Fi manager library to build all necessary pieces to use this feature. The benefits of this are enhanced mobility – Wi-Fi Direct devices can connect anytime, anywhere to each other directly since a Wi-Fi router of AP is not required – and ease of use, as there is no need for pre-configuration. All that users need to do is discover nearby devices, then send a connection request and secure connection. However, only the most recent devices support Wi-Fi Direct. The group owner can be a bottleneck, especially if the number of peers is high. Using this feature requires interaction between users: each user has to choose to whom they wish to connect and that chosen device has to accept the request before establishing communication.

An Australian group is working on a project called *Serval mesh* [9], [10] that aims to enable devices to perform self-organizing P2P networks with gateways to the Internet when available. Users can send text messages either to a particular peer or to all peers (broadcasting) in the same area; calls to any peers using the VoIP technique can also be made. The purpose of this project is to provide widespread communication in areas where the infrastructure is damaged, such as in the case of a natural disaster. A custom routing protocol that behaves in a similar fashion to the OLSR (optimized link state routing) protocol has also been used [11]. An Android app is available on the Android Market [12] which consists of two components: the Batphone and Serval DNA (distributed numbering architecture). The Batphone is the user interface; Serval DNA is the core networking, encryption and file-sharing component. The benefits of this project are that it is open source, free to use, carrier independent and all communication is done using Wi-Fi links. However, devices need to be rooted for full features. It works without rooting if peers are connected to the same network. The entire software is still in the experimental stage.

The *MANET Manager App* [13], which is part of a larger project called Smart Phone Ad-Hoc Networking (SPAN) [14], is made up of several components. These are the MANET service, modular MANET routing protocol framework, and a layer of managers for security, session management, and reliable transmission. It provides the choice of several reactive and proactive routing protocols through the modular routing protocol framework. The app components work together to provide a Java observer interfaces that other apps can implement. The apps can then interact with the network that the service creates.

By modifying the *Wi-Fi configuration file manually*, some developers have managed to make the kernel to get ad-hoc network working: the Wi-Fi chip is changed from its normal mode to an ad-hoc mode. In practice, users have to add ad-hoc network details (SSID, priority, ID) in the wpa_supplicant.conf file, and then set the ap_scan equal to 2 to perform ad-hoc networking. The benefit of this approach is its direct and easy way of allowing ad-hoc networking. However, mobile devices will either work in an ad-hoc mode or normal mode; the two modes cannot run at the same time. Moreover, it does not work on all Android devices.

III. System Design

A. System model

The MANET-cloud system is composed of two components with different characteristics: the public cloud and the ad-hoc network (see **Fig. 1**). The main features of a MANET are the mobility of its members and its short lifetime. Another important feature is that mobile devices are scarce in resources, especially in terms of energy and storage. In contrast, the cloud has extensive computing resources that can be allocated on demand and energy is not an issue. The communication between a MANET and the cloud can be provided either by cellular networks or Wi-Fi, allowing us to assume that at least one device in the network will be connected to the cloud at a time, all the time. The number of MANETs connected to the same cloud for the purpose of using cloud services is variable and can be very high in dense areas.

The new cloud service that we propose is MANET management. The role of this service is to manage all registered MANETs individually. When a MANET is created, it will register with the cloud, and, consequently, a new instance of the MANET management cloud service will be created and allocated to it. This service instance will monitor MANET membership and the communication link status between individual MANET devices and the cloud. In this way, the cloud service can select the best communication link with each MANET or switch from a link with a device that is departing to the best quality link among the remaining devices. One important consequence of this model is that in areas with a large density of mobile devices (phones), the number of data communication links can be decreased and, therefore, reduce the overcrowding of the radio spectrum.

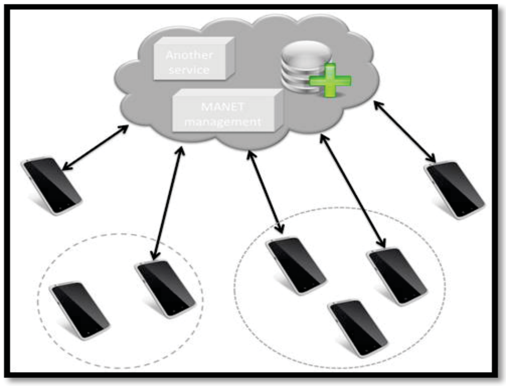

Fig. 1 The MANET and cloud model

The cloud service allocates identifiers to MANETs and IPs to joining members. This centralized management of addresses avoids the expensive operation of checking for duplicates whenever a new device joins the MANET or when networks merge. Additionally, the MANET management cloud service can store MANET information on a constant basis.

B. MANET management function

In order to avail themselves of the service, each user who has an Internet connection and is able to reach the cloud is required to create an account in the cloud by providing a unique username and a password.

When creating an account, the request is received into the cloud and its data checked against the users' database (DB). If the details are found in the DB, the cloud returns a MANET SSID to the user's mobile, as well as an allocated IP address. The MANET SSID is created based on user location and the capacity of each MANET. The IP address is allocated by a DHCP (Dynamic Host Configuration Protocol) server running in the cloud.

When the user details are validated, a profile table is created in the mobile device database, which contains account information such as: username, password, MANET_IP and MANET_SSID. A number of services are provided to the user, such as starting and stopping ad-hoc networking, sharing a file with a neighbour or all neighbours (broadcasting), and receiving notification from the cloud.

1) Starting-up a MANET protocol

A user starts a MANET by choosing the create MANET service, where a MANET SSID is taken from the profile table in the mobile DB and an IP address is set up statically from the profile table. When the MANET starts successfully, a hello message is periodically broadcast. The device also listens for any traffic in its radio range.

2) Joining an existing MANET

The procedure for joining an existing network depends on the availability of the cloud.

a. The cloud is unavailable

If the cloud cannot be reached, the user's device can be configured to a global MANET and the IP address is set statically from the device's MAC (media access control) address to avoid conflict (self-allocation). Global MANET is an open MANET network where some members have direct connection to the cloud and some do not. The idea behind this network is to provide a kind of connectivity when users cannot reach the cloud directly. Members in this network can use neighbours' links to communicate with the cloud. Similarly, the cloud can communicate with those members who do not have an Internet connection through those who have an active link with the cloud.

When a member acquires the ability to communicate with the cloud directly, one of the following scenarios will occur. The member verifies its information with the cloud and stays in the global network, or leaves the global network and either creates a new network or joins an existing one.

b. The cloud is available

When a mobile device is able to reach the cloud, it will successfully login to its account. It can then issue a request to join an existing MANET. The cloud replies by including all available MANETs in its vicinity that are found in the database. After the requester chooses a network from the list, the cloud allocates resources such as CPU, memory, storage, and networking resources and an IP address to be used in the chosen network.

3) Monitoring MANET activity

The cloud tracks each MANET usage and its members' behaviour for the purpose of providing better performance. For example, the cloud can switch from one link to another to communicate with a particular member. By gathering information about network activity, the cloud can balance networks such as by adding more members or redirecting them to another network to avoid overcrowding when other networks are scarce or inactive. Indeed, all information can be analysed to provide statistics for the service provider service provider.

However, each member of a MANET can periodically send a message to the cloud for identification of its status and confirmation of the network to which it is connected. Alternatively, to save energy, a member will contact the cloud only when its status is about to change or has changed – departing/departed. On the other side, the cloud has to update the users' records when status messages are received as well as check inactive users by sending a 'check status' message.

a. Cloud agent

For the purpose of tracking each MANET efficiently and providing services to members who do not have an active link, the cloud chooses one of the MANET members as a cloud agent. The main role of this agent is to feed the cloud information in regard to its connected MANET, such as MANET size and members joining/departing. The cloud can also reach members through this agent as well as being able to broadcast messages to the whole network. Initially, the cloud picks the issuer of a new MANET as a cloud agent of the MANET. Then, if other members start to join the MANET, the cloud can change the MANET agent if there is another node that has a better link. However, if the agent has left the network, the cloud will start to check the status of all links where the node with the best link with the cloud will have a higher chance of being selected. After choosing a MANET agent, the cloud starts to retrieve MANET information from that agent.

b. MANET split operation

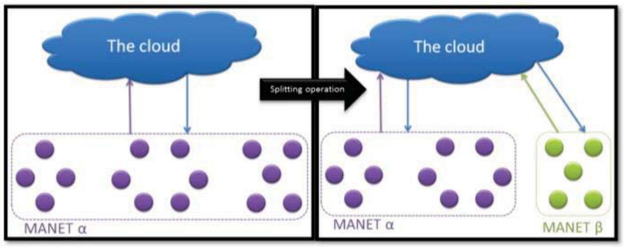

Fig. 2 The cloud splitting a MANET into two separate networks

For the purpose of providing better performance to users, the cloud will intelligently perform actions based on statistics gathered from monitoring MANET behaviour. One of these actions is a split, which is separating an existing MANET into two (or more) networks.

The cloud can split an existing network based on certain parameters, such as MANET size, the purpose of creating/joining this network, and network activities. For instance, if a network is found to be congested and overcrowded, a split operation is practical. Another example of implementing a split operation is to group members according to some criterion, e.g., role – doctors, nurses, firefighters and so on.

The cloud starts the split operation by listing users who will move to a new network. It then distributes the new MANET SSID to those users where the IP addresses are kept as they have already been allocated from the cloud. When users receive this announcement from the cloud, they disconnect from the old network. **Fig. 2** is an illustration of how the cloud splits a network into two separate ones where the old SSID is kept.

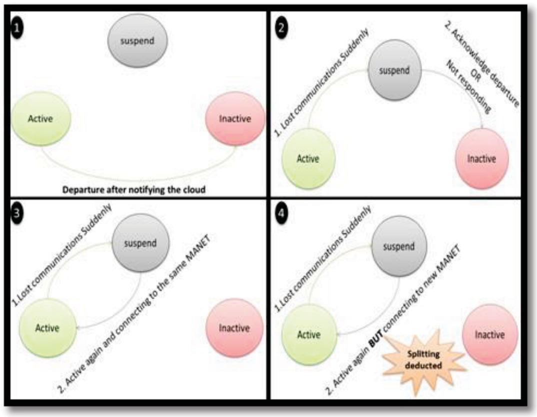
1
suspend
Active
Inactive
Departure after notifying the cloud
2
1. Lost communications Suddenly
suspend
2. Acknowledge departure OR Not responding
Active
Inactive
3
1.Lost communications Suddenly
suspend
2. Active again and connecting to the same MANET
Active
Inactive
4
1.Lost communications Suddenly
suspend
2. Active again BUT connecting to new MANET
Active
Splitting deducted
Inactive

Fig. 3 Users' status

However, this operation can also be started by mobile nodes when the departure of one (or more) node(s) is detected. The cloud detects that the network is about to split in different ways, such as: fading signals from some MANET members, not receiving anything from some members, or retrieving network operations via the cloud agent. In practice, the cloud marks users' status to one of three categories, namely, active, suspended, and inactive. Active users send and receive messages using the cloud. In contrast, inactive users have left their MANET after notifying the cloud of their departure. Between these two types, there are suspended users, where communication from/to those users suddenly stops without any previous notice. Losing communication between the cloud and such a user can be the result of different circumstances, such as the mobile devices (battery dead), networking issues (no Internet access), or connecting to a MANET that is not registered in the cloud. Therefore, the cloud has continuously (every ten minutes for example) to try to reach suspended users to update their status by sending a direct message to that member or retrieving its status from the cloud agent of its MANET. One possibility is that a suspended user was connected to MANET α and it is found that it is now connected to a new MANET β, which is not registered in the cloud database. The cloud then checks if there are other members of MANET α that are now members of MANET β. If that is the case, the cloud considers that MANET α has split into two networks: one which has the same SSID and a second one that has a new SSID, i.e., β. Finally, the cloud starts monitoring the new MANET to capture its user behaviour as well as to allocate resources to this MANET. Fig. 3 provides examples of users' status where, as in the last example, splitting is detected by the cloud.

c. MANET merge operation

Essentially, managing a high number of MANETs will affect system performance. Therefore, the cloud has to ensure, first, that creating a MANET is not being used as an attack to harm the whole system. The cloud then has to reduce the number of MANETs to provide a reliable service to users. One of the models for reducing MANET numbers is by merging two (or more) MANETs to a single MANET. The merge operation can be performed by the cloud when needed. If two small MANETs are located in the same area, merging these will be beneficial.

To ensure that merging is accomplished efficiently, the cloud keeps IP addresses of both networks' members. The SSID of one of these networks will be used for the new network. Choosing which SSID will be used depends on the size of each network. In other words, the SSID of the MANET that has the higher number of members will be used. However, merging can occur between users without involving the cloud. If that is the case, the cloud has to have a mechanism for detecting this event. For example, users on a train might create a new MANET (called α) to share data and they arrive in another city that has a number of MANETs, where most of those users start joining a particular MANET (called β). The merge has to be detected by the cloud for a number of purposes, such as the allocation of extra resources. In practice, the cloud watches members' activities (joining/departure) that are sent by the cloud agent mentioned previously. If the cloud finds that a number of members that are connected to another MANET have started to join the agent's network, the cloud assumes a merge operation is in progress. When merging is detected, the cloud releases resources allocated to the old network and allocates more resources to the MANET that is left. In the case that the old network is not known to the cloud, a record of what has been done will be created in the cloud database. **Fig. 4** shows an example of the merge operation.

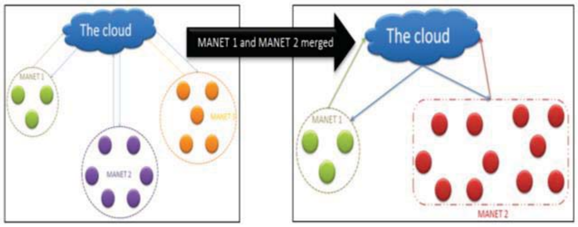

Fig. 4 Merging operation performed by the cloud

d. Session still active

One of the most important benefits of introducing cloud services is saving an active session, even if one of the two parties has left the network. There can be nodes in a client-server relationship and they can be engaged in active sessions. If, due to mobility, the communication links are broken, the sessions are affected. In a traditional MANET, if two nodes are communicating to do a job and one of them departs suddenly, the communication is aborted. The session is not active anymore and all work is lost. However, in our system, when two nodes registered in the cloud have an active session and one of them is unexpectedly disconnected, the session can be saved. The link break will determine the lack of acknowledgement from the departed node. The member of the MANET will start the session-saving procedure in the cloud, assuming the session is saved on the receiver side also. It will then return to being active when the sender sends a request to the cloud looking for the disconnected node to resume that job. The cloud has the role of acting as a bridge between these two nodes to keep the session active (see **Fig. 5**).

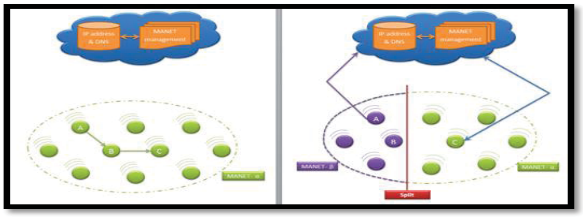

Fig. 5 Split has occurred and session is still active

IV. Implementation

The system was implemented using Android Java SDK [15] and tested on three Google Nexus One (running Android 2.3) and two Samsung Galaxy III (running Android 4.2.2) mobile devices. Super-user access was provided to allow Wi-Fi chip access and modification. Some devices had a 3G link and some did not.

Cloud services were implemented using Amazon Elastic Compute Cloud (EC2) [16], where the instance was deployed in the EU (Ireland) region. Amazon Simple Storage Service (S3) [17] was used for storage purposes. When users communicate with the cloud to seek services, their information is stored in a database that is hosted on the cloud using the Amazon Relational Database Service (Amazon RDS) [18]. The database contains the following:

- **Users Account Table:** contains details of all users.
- **Actions Table:** contains all users' requests sent to the cloud.
- **Broadcasting Table:** stores details of each broadcast message that is sent from the cloud to users.
- **Neighbour Table:** stores ad-hoc network members and actions.

As is known, administrators can manage Amazon services in different ways, such as accessing an AWS (Amazon Web Services) console. Alternatively, we developed another Android app (on Samsung Tab 2) for administrator use for sending notifications, checking users' status and the retrieval of statistics about system usage. A website is also hosted in the cloud to ease interactions with cloud services. Furthermore, an Android app was developed and installed on mobile devices to provide the MANET services described in the following sub-sections.

A. Account management

Users are required to login using an account that is created in the cloud. After providing a username and password, the app sends these data to the cloud to be validated. The cloud checks if the user exists and if the information is correct. When the user is logged in successfully, a profile table is created in the mobile device database.

B. Receiving notifications from the cloud

Clients connect to the cloud to seek services. The cloud answers requests after it stores them in the database. Sometimes, however, the cloud service needs to notify users by broadcasting an alert. The cloud can push information to a group of devices without involving them in the operation. In practice, there are different ways to do so, such as sending GET requests periodically to the cloud to pull messages.

However, we are using another solution that is provided by Amazon called the Amazon Simple Notification Service (Amazon SNS) [19]. In SNS, developers can send notifications (messages, email, or SMS) to endpoint users using one of the existing platforms [20],[21]. Since this implementation is deployed on Android-based devices, Google Cloud Messaging (GCM) for an Android platform is used [21].

When a user logs in successfully the app registers on the GCM server and then sends the ID to the cloud. In the cloud, IDs are stored in the database where an administrator can write a message and send it to users, either to a particular user or to all registered users. On the user side, notifications will be presented in the notification bar. By clicking on the notification bar, the content of the notification is expanded.

C. Mobile ad-hoc network service

Two ways of performing ad-hoc networking in Android mobile devices are used. First, the MANET app is used as a library to start and stop the MANET network [12]. Second, by modifying a Wi-Fi configuration file, as explained below.

1) Starting MANET

The app will execute the following steps to start the ad-hoc network:

- A copy of the current wpa_supplicant.conf file is saved in the SD-card as a back-up.
- MANET details are added to that file and the ap_scan value is set to 2.
- The IP address is set up statically, which is taken from a profile table.

2) Stopping MANET

In order to stop the ad-hoc network of mobile devices, the app executes the following steps:

- All MANET details are deleted from the wpa_supplicant.conf file and the ap_scan value is set back to 1.
- The IP address is set back dynamically.
- The neighbour discovery service is stopped.

If the ad-hoc mode is established successfully, the neighbour discovery service is started. Each device broadcasts a beacon every second (HELLO packet) using the UDP (User Datagram Protocol) to reduce network overhead, while listening to any packets in the same radio range.

D. Sending messages service

To test the feasibility of the aforementioned system, a message sending service is offered where users can send a message to neighbours either directly or multi-hop. Users can also send messages to the cloud directly or using another peer's link. The time needed for a message to reach its destination is calculated on the sender side.

E. Sharing files service

Besides the previous service, file sharing is also provided. Users can send a file to a peer either directly or via routing. The cloud can also be involved in sending files between two nodes that were connected to the same network but which then departed.

F. Communication types

Both services are evaluated using different communication types, presented as follows.

1) Communication between MANET members directly

When a destination is chosen, a Wi-Fi link is established using the TCP (Transmission Control Protocol) to ensure delivery. On the receiver side, a notification is triggered to notify the receiver about receiving new data. If sent data contain a file, then it will be stored in the device's SD-card.

2) Communication within MANET (multi-hop)

To achieve multi-hop communication, the sender is forced to route the request to one of its neighbours found in the neighbours' list to reach the receiver – The routing table.

In the app, users either have to choose a file from the SD-card, or type in a message to be sent. Then a destination from the neighbours' list is chosen. Afterwards, the file or text message is sent to another neighbour, including the details of the destination node. When the data arrive at the intermediate node, a notification is generated to notify that node of being involved in sharing data between the source and the destination. If the destination node receives the data successfully, the source node will be notified by the intermediate node.

3) Communication to the cloud (directly)

If a sharing file service is used, then a multi-part http request is built containing file details to be uploaded to the cloud directly using a Wi-Fi or 3G link. When the file arrives at the cloud, it will be stored on S3 space. If the messaging service is chosen, the message's text is sent as a direct http request to the cloud. When the cloud receives the message, an acknowledgement will be sent to the sender.

In both services, the sender calculates the duration of the whole operation to determine if the cost resulting from using each service is reasonable and meets users' expectations.

4) Communication to the cloud via neighbours' links

If such a user could not reach the cloud directly, a neighbour's link can be used. Simply, files/messages are sent directly to one of the neighbours found in the Neighbour Table, where the chosen neighbour acts as an intermediate node between the sender and the cloud to transfer the request. This intermediate node will also send an acknowledgement to the sender when data are delivered to the cloud successfully. In this way, the sender will be able to compute the time resulting in sending these data.

In a real-world scenario, a request routed from a sender to the cloud may travel over more than one intermediate node. Therefore, we force the request to be redirected via neighbours to achieve one-hop, two-hop, three-hop, and four-hop communication to test the ability of serving requests from a sender to the cloud and vice versa.

5) Communication to a neighbour via the cloud

By the nature of MANET, if a member leaves the network that means that messages can no longer be received. To benefit from introducing the cloud to our system, messages that are sent to a departing node can be redirected to the cloud to ensure delivery.

In the app, the sender has to choose a destination from *the recently-added-neighbour table*. This table contains a list of all neighbours who have been discovered since joining the current MANET. Subsequently, the request is sent to the cloud and includes the destination details (name, IP address). Therefore, the cloud matches these details with the receiver's current GCM ID before redirecting the request. Furthermore, the cloud checks the status of the destination node, whereby the request is scheduled to be delivered later if the destination node is not connected. When the request is received by the receiver, the cloud will send an acknowledgement to the sender.

V. EXPERIMENTAL RESULTS

This section presents the performance results of our system implementation. Each mobile device was placed away from the others but within radio range. Performance data were collected for both communications between devices inside ad-hoc networks, including direct and multi-hop communications and between the cloud and mobile devices. Each result is the average of experiments having been conducted ten times.

A. Sharing file results

A small text file (1 KB) was transferred over TCP from the sender to the receiver. The round-trip delay time (RTD) was calculated at the sender by comparing the system current time when the request was sent with the system current time when the acknowledgement was received.

TABLE 1. EXPERIMENTAL RESULT OF THE SHARING FILES SERVICE

	Communications type	***Avg. (RTD)***	
To the cloud	Directly using Wi-Fi	0.447	**Seconds**
	Directly using 3G	1.7111	
	Via a neighbour's link	4.3647	
To a peer	Directly	2.0508	
	One-hop	4.2933	

Table 1 shows that the time needed to share a file directly between two nodes took around 2 seconds. It took almost double that time when the file was sent via another neighbour. This result is explained by the fact that a copy of the file is stored in the intermediate node. Interestingly, sending files over the cloud took almost the same amount of time as sending files by two-hop communication. The length of time will be affected by some factors such as availability of the destination node, and the quality of connection between the cloud and the mobile devices.

B. Sending messages service results

A string containing "hello" was sent over TCP from source to destination. The delay in receiving this message at the destination was calculated on the sender side. The following table presents the results of this service using different communications types.

TABLE 2. EXPERIMENTAL RESULT OF SHARING MESSAGES SERVICE

		Communications type	***Avg. delay***	
Sending messages	**To the cloud**	Directly using Wi-Fi	260.3	**Milliseconds**
		Directly using 3G	2825.2	
		Via neighbour link (one-hop)	2701.6	
		Via neighbour link (two-hop)	2776	
		Via neighbour link (three-hop)	2883.2	
		Via neighbour link (four-hop)	2921.1	
	To a peer	Over the cloud	3864.4	
		Directly	34.9	
		One-hop	63.6	

Table 2 indicates that the time needed to send a message directly to the cloud using a 3G link was almost the same as the time taken to send a message to the cloud via neighbours. Furthermore, the delay might be less if another neighbour has a better link. There are two main reasons for this result: firstly, the delay time added by communicating inside the MANET is very small compared to the delay that results from communicating with the cloud. Secondly, the quality of links varies, which means some nodes have better links than others. This means that communicating with the cloud via a neighbour who has a better quality link will reduce the overall delay time. Furthermore, the delay time can be less than when sending the same request over a low-quality link.

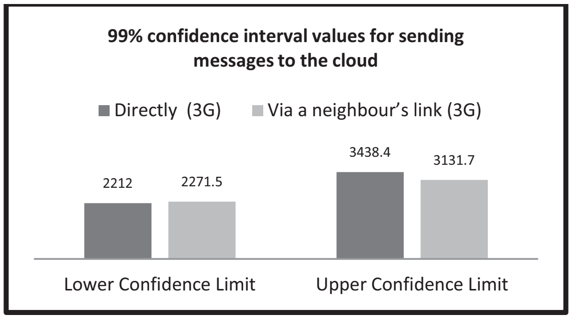

Fig. 6. Chart comparing the 99% confidence interval values of sending messages directly to the cloud and sending the same messages via a neighbour's link.

However, sending messages between two peers via the cloud takes around 4 seconds, because using the aforementioned notification system (SNS with GCM) adds an extra delay to the request, whereby the request travels from the sender to the cloud and then to the GCM platform before reaching the destination node. The acknowledgement will also travel all the way back to notify the sender.

VI. Conclusion and Future Work

This paper has presented a new model for a mobile ad-hoc network management that introduces cloud services. The design and implementation of the system have been discussed. The most important benefits of this solution are its impact on MANET operations such as split, merge, join and leave. The system is reliable and robust, as split, for example, does not require re-allocation of IP and SSID addresses, as they are managed by the cloud. All local communication between mobile devices for split management becomes unnecessary, and, therefore, saves mobile resources. The same applies to merge, join and leave. We also presented ways of creating ad-hoc networking on Android devices.

Experimental results show that the proposed model is feasible and produces excellent results under lab conditions. According to **Fig. 6**, we are 99% confident that the mean delay resulting from sending messages to the cloud directly and the delay resulting from using a neighbour's link are very similar. Furthermore, the delay could be lower if that neighbour has a better link to the cloud.

In conclusion, managing a mobile ad-hoc network of Android-based devices by the cloud is achievable and offers features such as reliability and robustness.

As future work, we are planning to enhance the MANET management service in the cloud, where it could switch smartly between active-links, and allocate/de-allocate resources. Additionally, testing the proposed model in a real environment where different levels of mobility will be considered, as well as more cloud services to be shared by MANET members, will be deployed.

VII. Acknowledgements

Hazzaa Alshareef's PhD research is funded by the King Abdullah bin Abdulaziz Scholarship Programme (KASP), which is managed by the Ministry of Higher Education in Saudi Arabia.

VIII. References

[1] D. Grigoras and M. Riordan, "Cost-effective mobile ad hoc networks management," *Futur. Gener. Comput. Syst.*, vol. 23, no. 8, pp. 990–996, Nov. 2007.

[2] S. Nesargi and R. Prakash, "MANETconf Configuration of Hosts in a Mobile Ad Hoc Network," *Proc. IEEE INFOCOM.*

[3] M. Conti and S. Giordano, "Mobile ad hoc networking: milestones, challenges, and new research directions," *Commun. Mag. IEEE*, January, pp. 85–96, 2014.

[4] L. Zhang, X. Ding, Z. Wan, M. Gu, and X. Li, "WiFace: a secure geosocial networking system using WiFi-based multi-hop MANET," *Proc. 1st ACM Work. Mob. Cloud Comput. Serv. Soc. Networks Beyond.*, ACM, 2010.

[5] D. H. D. Huang, X. Z. X. Zhang, M. K. M. Kang, and J. L. J. Luo, *MobiCloud: Building Secure Cloud Framework for Mobile Computing and Communication*. Ieee, 2010, pp. 27–34.

[6] *Wi-Fi Direct^{TM} | Wi-Fi Alliance,Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.2*. Wi-fi.org, 2010.

[7] "SoftAP - Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/SoftAP. [Accessed: 23-Feb-2014].

[8] "Welcome to the Freedom of Portable Wi-Fi; Intel® My WiFi Technology: Synch, Share, Show & Print on the Go," 2008.

[9] P. Gardner-Stephen, "The serval project: Practical wireless ad-hoc mobile telecommunications," June, pp. 1–29, 2001.

[10] P. Gardner-Stephen and S. Palaniswamy, "Serval mesh software-WiFi multi model management," *Proc. 1st Int. Conf. Wirel. Technol. Humanit. Reli. - ACWR '11*, p. 71, 2011.

[11] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," *Proceedings. IEEE Int. Multi Top. Conf. 2001. IEEE INMIC 2001. Technol. 21st Century.*, pp. 62–68.

[12] "The Serval Mesh - Android Apps on Google Play." [Online]. Available: https://play.google.com/store/apps/details?id=org.servalproject. [Accessed: 05-Jan-2014].

[13] "MANET Manager - Android Apps on Google Play." [Online]. Available: https://play.google.com/store/apps/details?id=org.span. [Accessed: 15-Jan-2014].

[14] J. Thomas, J. Robble, and N. Modly, "Off Grid communications with Android Meshing the mobile world," *Homel. Secur. (HST), 2012 …*, pp. 401–405, Nov. 2012.

[15] "Android SDK | Android Developers." [Online]. Available: http://developer.android.com/sdk/index.html. [Accessed: 23-Feb-2014].

[16] "AWS | Amazon Elastic Compute Cloud (EC2) – Scalable Cloud Servers." [Online]. Available: http://aws.amazon.com/ec2/. [Accessed: 21-Feb-2014].

[17] "AWS | Amazon Simple Storage Service (S3) – Cloud Storage." [Online]. Available: http://aws.amazon.com/s3/. [Accessed: 21-Feb-2014].

[18] "AWS | Amazon RDS - Cloud Relational Database Service." [Online]. Available: http://aws.amazon.com/rds/. [Accessed: 21-Feb-2014].

[19] "Amazon Simple Notification Service (SNS) Documentation." [Online]. Available: http://aws.amazon.com/documentation/sns/. [Accessed: 23-Feb-2014].

[20] "Local and Push Notification Programming Guide: Apple Push Notification Service." [Online]. Available: https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html. [Accessed: 23-Feb-2014].

[21] "Google Cloud Messaging for Android | Android Developers." [Online]. Available: http://developer.android.com/google/gcm/index.html. [Accessed: 23-Feb-2014].