



**HAL**  
open science

## Opportunistic information dissemination over hybrid ASN using publish/subscribe communication model

Martin Xavier Tchembe, Maurice Tchoupe Tchendji

### ► To cite this version:

Martin Xavier Tchembe, Maurice Tchoupe Tchendji. Opportunistic information dissemination over hybrid ASN using publish/subscribe communication model. 2022. hal-03585037v1

**HAL Id: hal-03585037**

**<https://hal.science/hal-03585037v1>**

Preprint submitted on 24 Feb 2022 (v1), last revised 4 May 2023 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Opportunistic information dissemination over hybrid ASN using publish/subscribe communication model

Martin Xavier Tchembe  
Maurice Tchoupe Tchendji

---

## Abstract

Ad hoc social networks (ASNs) are social networks that exist within a mobile ad hoc network (MANET). Nowadays, almost all mobile devices such as smartphones and laptops have the ability to connect directly to each other and thus form a MANET without the need for a pre-existing infrastructure. Communication in such a social network usually consists of a station broadcasting its interests to the underlying MANET. If these are strongly similar to those of another station nearby in the network, they can then engage in a conversation. However, the coverage of ASNs is limited to a small geographical area, due to the limited communication range of MANET stations. In this paper, we propose a protocol for opportunistic information dissemination in hybrid ASNs (ASNs whose underlying MANET is coupled to an infrastructure network). It extends the Social-MANET protocol on which it is based, and which is an information dissemination protocol for MANET stations using the topic-based publish/subscribe communication model. We also propose an implementation of the extended protocol in a prototype Android application for information dissemination in a scientific conference. The results obtained after this implementation allowed us to show the applicability of our solution.

## Keywords

Ad hoc Social Networks ; Publish/subscribe ; Topics ; Hybrid MANET ; Information dissemination ; Nearby Connections

---

## I INTRODUCTION

Ad hoc social networks (ASNs) are social networks that use mobile ad hoc networks for their implementation and to provide communication between different stakeholders. Sapna Gambhir in [8] defines ASN as a close network that uses the ad hoc network to socially connect interested users; a combination of a social network that maintains the profile and interests of the users, and an ad hoc network that helps to connect nearby users without a centralized access point. ASNs can be used in social locations such as travel stations, football stadiums, hospitals, school and university campuses, hypermarkets, fairs, etc. ASNs can be used in scientific conferences for example to allow participants to communicate and exchange information such as personal notes, the conference program, or even important information given by the moderators (in case a participant arrived late or was absent for a while for example). Nagender Aneja et al in [16] listed many other application areas of ASNs, including cellular network extension. Indeed ASNs can be deployed in areas not covered by infrastructure networks such as areas of natural disasters that have damaged the existing infrastructure.

Communication in ASNs is generally based on interests. A station broadcasts its interests into the network. These can be received by a neighbouring station. If a match is found between its local interests and those received, the two stations can then engage in a conversation as they share common interests. In [12] for example, an architecture and implementation of social networks on commercial mobile devices is presented. This implementation allows the name and a limited number of keywords representing users' interests to be broadcast without any connection in a nearby region in order to facilitate the connection of individuals. A similar principle is employed in [13] where stations in an ASN communicate via publish/subscribe by periodically broadcasting announcements containing the identifiers of the topics they have subscribed to. These topic identifiers are used to identify stations that share the same interests.

Thanks to the underlying MANET, ASNs can be set up spontaneously without the need for any pre-existing infrastructure and therefore have the advantage of being all-weather, all-terrain networks. They also have the advantage of being inexpensive and accessible to all. Indeed, almost all mobile devices (smartphones, laptops, etc.) nowadays have the capacity to communicate in an ad hoc manner thanks to technologies such as Bluetooth and Wi-Fi. These devices have become cheap and accessible to all. However, it is easy to see that such networks, although having the advantage of being cheap and accessible, are limited in terms of network coverage because by definition, the stations of such a network communicating by radio wave cannot do so beyond a certain distance, which limits the range of the network. If we take the example of the scientific conference mentioned above, let us imagine that it is held in several sites located in different cities, for example (the case of the African Conference on Research in Computer Science CARI'2022 with three sites, Tunis, Yaoundé and Dschang). An ASN could be used by the participants of one site. However, in order to extend communication among participants in different sites, the need to introduce an infrastructure network becomes obvious. The existence of a hybrid ad hoc social network in this case could allow participants from different conference sites to be connected as if they were all on the same site. A participant in the site of Yaoundé who has access to the infrastructure network (e.g. the Internet) could get an information and disseminate it to other participants in the same site who do not have access to this network.

In this paper, we propose a protocol for information dissemination within ASNs coupled with infrastructure networks. This coupling will not only extend the range of ASNs but also extend the coverage of the infrastructure network as stated by Mohammad Al Mojamed in [14]. The communication model we consider is the topic-based publish/subscribe communication model. Indeed, the protocol we propose in this paper is based on the SocialMANET protocol. This is a publish/subscribe information dissemination protocol for MANETs presented in [13], which we extended to make it able to work in an ASN whose underlying MANET is coupled to an infrastructure network. The use of a publish/subscribe communication model allows targeted dissemination of information. Indeed, this communication model suggests that each station subscribe to a certain number of topics, thus defining its interests and hence the types of information it wish to receive. Stations can then be notified when information belonging to one of their interests is available. We also propose in this paper an implementation of the extended protocol within a prototype Android application for the dissemination of information among participants of a scientific conference.

The rest of this paper is organised as follows: section II presents similar works found in the literature, section III briefly presents the information dissemination protocol we used and section IV is dedicated to the presentation of the improvements we added to that protocol. In section V, we present an implementation of our solution within a prototype application and the results

obtained in section VI. Section VII concludes this paper.

## II RELATED WORKS

In [9], the authors clearly illustrated the need to couple ASNs with infrastructure networks for better accessibility to the information disseminated and for a wider coverage of both ASNs and infrastructure networks. In a case study that the authors presented concerning vehicular social networks, they identified three types of communication: vehicle-to-vehicle, vehicle-to-person (pedestrian) and vehicle-to-infrastructure.

Efforts have been made in the literature to provide solutions to the problem of integration of ASNs and infrastructure networks. In [2], the authors present the challenges related to this integration and review the strategies used in the literature. The solutions proposed in [14], [3] and [11] consist in extending a routing protocol used in MANETs to provide bi-directional connectivity between ad hoc stations and hosts in infrastructure networks. The common principle of these solutions is to use a set of ad hoc hosts known as Gateways to act as bridges between the two networks.

In [17], the authors propose a secure publish/subscribe system based on smart contract in autonomous vehicle networks using vehicles which are part of a truck platoon system as brokers of the publish/subscribe system. A cloud server is used to establish a contract between subscribers and brokers.

The authors in [4] propose an information dissemination protocol based on content-based publish/subscribe in a hybrid VANET. The protocol uses the infrastructure (consisting of a set of fixed information stations distributed in the city and able to communicate with vehicles via Wifi) if existing, and vehicle-to-vehicle communications. A published message is disseminated within a specific geographical area called the home zone (e.g. the area surrounding an accident point). Vehicle-to-vehicle and vehicle-to-infrastructure network communications are used to maintain publications within their home zones. Although this approach is effective as shown by the simulation results presented by the authors, it is highly dependent on GPS, navigation system, digital city map, etc. and requires very sophisticated vehicles equipped with these technologies. This is the observation made in [5] where the authors also present a publish/subscribe system for hybrid VANET environments consisting of fixed information stations and mobile vehicles. In their approach, the information stations are connected in DHT mode (Chord [1]) to the Internet and act as rendezvous points for publications and subscriptions. These information stations are also equipped with an omnidirectional IEEE 802.11 antenna and have a fixed range as much as the vehicles. Vehicles can therefore communicate in ad hoc mode with the information stations in the same way as they communicate with each other. Communication among vehicles and from vehicle to information stations is done by flooding, which can be a real bottleneck and lead to heavy power losses of the network stations. Moreover, while vehicles are not greatly affected by the limited energy constraint characteristic of MANETs in general, many devices such as mobile phones that could constitute a MANET network still suffer from this limitation, making it difficult to generalise this solution to ASNs.

### III INFORMATION DISSEMINATION OVER MANET STATIONS: SOCIALMANET

This section gives a brief description of SocialMANET, a protocol for information dissemination over MANET. For more information on it, interested reader should rely on [13].

SocialMANET is an information dissemination protocol using a topic-based publish/subscribe communication model in an ASN; it was proposed by Maurice Tchoupé et al [13]. It is subdivided into three sub-protocols which are: (1) the subscription sub-protocol, (2) the publication sub-protocol and (3) the dissemination sub-protocol. The first two give a hint of what they consist of by their rather indicative names and are relatively simple. The third sub-protocol is the core of this protocol and allows for the dissemination of publications from one network station to other stations based on their interests and/or altruistic nature. It consists of two phases carried out by all stations in the network independently of each other: the *needs detection phase* and the *publications dissemination phase*.

SocialMANET stations periodically broadcast an announcement message named *QUERY* (format illustrated in Figure 1) during the *needs detection phase*. This message contains the identifiers of the topics they have subscribed to, and for each of them the identifiers of the publications they have on these topics. The reception of this message triggers the *publications dissemination phase* at the receiving stations. During this phase, the information contained in the *QUERY* message is used by the receiving stations to calculate which publications they can provide to the sending station, according to their common interests in terms of topics they have subscribed to. Once this calculation has been made, and if they find that they store at least one such publication, they broadcast the publications they own according to the recent calculation. When a station receives a publication on a topic it has subscribed to and does not yet own, it stores it and is therefore able to broadcast it as well.

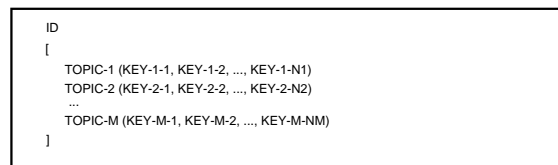


Figure 1: QUERY message format

SocialMANET uses an opportunistic approach whereby a station only disseminates information during fugitive contacts with other interested stations. This is to avoid wasting the stations' energy on disseminating information unnecessarily or having them participate in the dissemination of information that they are not interested in. In practice, this protocol is run on a virtual network layer within an existing physical network (a mobile ad-hoc network). Any station in the network participating in the execution of the SocialMANET protocol is autonomous and is able to communicate with any other station in its neighbourhood (a station's neighbourhood consists of the stations that are within its radio range). Each station can be both a publisher and a subscriber. All stations can disseminate the publications they own to interested subscribers.

### IV SOCIALMANET REVISITED PROTOCOL

Because of the limitation due to the short communication range of the stations in a MANET, the information exchanged in an ASN is hardly accessible by very distant stations. The question of the need to couple MANETs to infrastructure networks to overcome this difficulty has been

addressed by several authors in the literature. The solution proposed in this paper is based on the protocol for information dissemination in ASN, SocialMANET. It is an extension of SocialMANET, with the aim of making it able to support integration with infrastructure-based networks. Our proposal is methodologically very similar to the solutions presented in [3], [11] and [14]. However, while in [3], [11] and [14] authors extend a routing protocol used in MANETs to provide connectivity between ad hoc stations and hosts in infrastructure networks, we rather extend the communication protocol in order to provide this connectivity. This extension is perceptible at two levels. Firstly, at the level of the structure of the network considered, with the integration of an infrastructure network in which we have fixed stations that can be contacted by mobile stations and communicate with them. Secondly, at the level of the dissemination sub-protocol of the SocialMANET protocol, with the introduction of opportunistic exchanges (exchanges initiated by mobile stations when they have access to the infrastructure network) between the mobile stations of the MANET and the fixed stations of the infrastructure network.

#### 4.1 Network Design

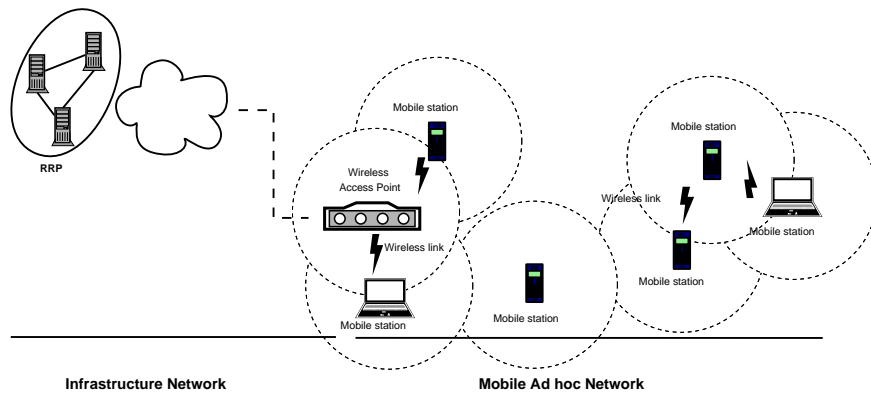


Figure 2: Network architecture

Figure 2 shows the general architecture of the network we are considering. We have mobile stations that communicate with each other via radio waves, using technologies such as Bluetooth or WiFi. The infrastructure network can be Internet or any IP network with fixed and perfectly addressable stations. The communication between the MANETs mobile stations and those of the infrastructure network can be done through the cellular network (3G/4G) or through a Wifi access point for example. Each MANET station is able to contact a station on the infrastructure network if it has access to it. In the case of an Android smartphone, for example, access to the infrastructure network can be detected by a change in the status of the Wi-Fi indicating the connection to an access point when the Wi-Fi is activated, or when the user activates the mobile data connection (3G/4G) for example.

No MANET station acts as a gateway for other stations to access the infrastructure network. It is true that several solutions in the literature [3, 11, 14] employ this technique to allow even MANET stations without direct access to the infrastructure network to access it through other stations (gateways). However, this places an additional load on the gateway that acts as a relay between stations on the infrastructure network and those on the MANET that do not have access to the infrastructure network. This load is likely to lead to additional energy consumption of the gateway, which is why we avoid using mobile stations as gateways for other mobile stations. This choice also adheres to the principle of non-waste of energy used by SocialMANET [13],

which consists of a station avoiding wasting energy to participate in the dissemination of messages that are not intended for it or for which it has no interest, in order to avoid running out of energy very quickly.

In the infrastructure network, we consider the existence of an entity called Remote Rendezvous Point (RRP). It can be station known and accessible by all MANET stations having access to the infrastructure network. It is considered to be subscribed to all topics in the publish/subscribe system. It is considered to have an unlimited amount of storage space and energy, so that it can store all publications coming from any MANET station. The RRP can also be a set of fixed stations collaborating in a distributed way to ensure the storage and dissemination of publications, and linked together by a Chord [1] network over Internet for example.

#### **4.2 Extension of the SocialMANET dissemination sub-protocol: Contacting the Remote Rendezvous Point (RRP)**

When an ASN station discovers that it has access to an infrastructure network, it instantly initiates a communication with the RRP located on this network by sending a message to it, identical to the classic *QUERY* announcement message of the SocialMANET protocol. Processing this message allows the RRP to deduce two pieces of information: (1) the identifiers of the publications to be sent to the ASN station and (2) the identifiers of the publications to be retrieved from the ASN station. In response, the RRP sends back to the ASN station the list of publications which it knows that it needs, after interpretation of the *QUERY* message received. It also sends the identifiers of the publications to be requested. The ASN station then sends these publications to the RRP and thus the RRP and the ASN station are all up to date with each other.

Our protocol is opportunistic for two reasons: First, it inherits the opportunistic nature of the SocialMANET protocol, which it extends. The opportunistic nature of SocialMANET consists in exploiting fugitive contacts among stations sharing the same interests to disseminate publications. Secondly, our protocol does not rely strongly on the infrastructure network for the dissemination of publications, but takes advantage of the temporal access of mobile stations to infrastructure networks when this occurs, to update itself and hence the RRP, regarding missing publications.

## **V ARCHITECTURE AND IMPLEMENTATION**

In this section, we present an implementation of our solution within a prototype mobile application for the Android platform called *ConfInfo*. The goal of the application is the sharing of information within an ASN formed by the participants of a scientific conference. The information to be disseminated is grouped into topics. Topics' names reflect the categories in which the participants of the conference can be classified: speakers (plenary speaker, session speaker, and talker), the organizing committee (secretariat, protocol, and logistics), chairs (plenary chair, session chair), guests, etc. Users have to subscribe to a given topic in order to receive related publications.

We simulate device-to-device connectivity in ad hoc mode using Google's Nearby Connections API. Indeed, although direct device-to-device connectivity via ad hoc mode is available in the IEEE 802.11 standard, it is not yet supported by Android. Researchers in the literature simulate it by using Wifi Direct or Wifi P2P which allows devices to connect to each other and form

groups [7, 10, 12]. Nearby Connections (<https://developers.google.com/nearby/connections/>) follows the same logic and offers interfaces that facilitate the discovery of nearby devices and communication with them.

## 5.1 Architecture

The architecture we have adopted is the one defined in [15]. Contrary to other architectures proposed in the literature, such as those presented in [6, 10, 12], it is perfectly adapted to ASNs in which stations communicate by the publish/subscribe communication model. The other architectures, on the other hand, are designed for ASNs where stations communicate synchronously after having discovered a high match rate during a comparison among their interests. The three

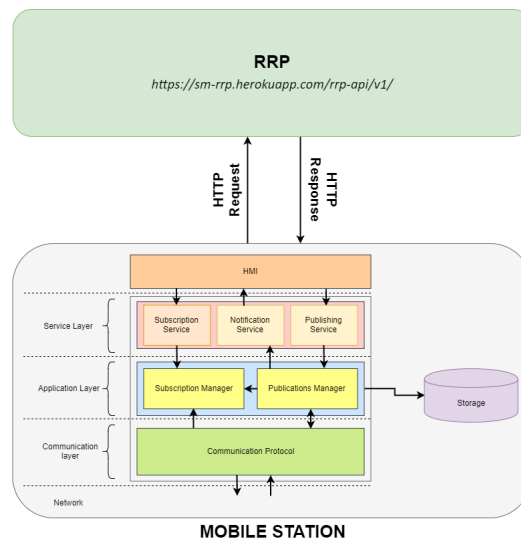


Figure 3: Extended architecture of SocialMANETs

main layers of this architecture are the following:

1. **Communication Layer**: it is responsible of functions such as discovering and updating information about neighbouring stations, and communicating with them;
2. **Application Layer**: through its Publications Manager and Subscription Manager components, it manages the user's topics, publications and subscriptions;
3. **Service Layer**: it offers users a set of services allowing them to subscribe to a topic, to publish information and even to be notified when a new publication is received from a neighbouring station.

We slightly extended this architecture by adding the **RRP** component, representing the remote and fixed entity located on the infrastructure network, able to store all publications and able to communicate with all mobile stations that have access to it (Fig. 3).

## 5.2 Discovering neighbouring devices

The *Nearby Connections* API of Google allows to advertise, discover and connect nearby devices in a completely off-line peer-to-peer manner. Connections among devices are high-bandwidth, low-latency and fully encrypted to enable fast and secure data transfers. Under the hood, the API uses a combination of Bluetooth, BLE and Wifi hotspots, taking advantage of the strengths of each while working around their respective weaknesses. For convenience, users are



not prompted to enable Bluetooth or Wifi. Nearby Connections activates these features as they are needed and returns the device to its previous state once the application has finished using the API.

Nearby Connections uses three advertisement and discovery policies. The *P2P\_POINT\_TO\_POINT* policy supports a 1-to-1 connection topology; it does not allow more than one connection at a time. The *P2P\_STAR* policy supports a 1-to-N, or star, connection topology. The *P2P\_CLUSTER* strategy supports an M-to-N, or cluster, connection topology. We used the latter because it allows devices to accept multiple incoming connections and simultaneously initiate multiple outgoing connections to other neighbouring devices.

The Java functions shown in the algorithms [V.1](#) and [V.2](#) allow a given device to play announcements to signal its presence and to start discovering nearby devices, respectively.

---

**Algorithm V.1** Advertising

---

```
1 public void startAdvertising ( ConnectionLifecycleCallback callback ) {
2     String UserID = getUserNickName(activity . getApplicationContext ());
3     Nearby.getConnectionsClient ( activity ). startAdvertising ( UserID,
4         NearbyConfig.SERVICE_ID, callback, NearbyConfig.getAdvertisingOptions ())
5         .addOnSuccessListener(new OnSuccessListener<Void>() {
6             @Override
7             public void onSuccess(Void aVoid) {
8                 activity . handleAdvertisingStatus ( true );
9             }
10        })
11        .addOnFailureListener (new OnFailureListener () {
12            @Override
13            public void onFailure (@NonNull Exception e) {
14                activity . handleAdvertisingStatus ( false );
15            }
16        });
17 }
```

---

The latter function accepts a callback that notifies the application of the presence of a nearby device. Once a device is detected, it is added to the list of nearby devices. The function *connect* (Algorithm [V.3](#)) is used to establish a connection with a nearby device.

Once the devices are connected in this way, they can communicate. The information transmitted is encapsulated in the Java class *com.google.android.gms.nearby.connection.Payload*. Three types of payloads are available:

- **BYTE**, allow to send simple data such as text messages,
- **FILE**, allow to send files such as photos and videos,
- **STREAM**, suitable for large amounts of data generated on the fly, such as an audio stream.

The function *sendPayload* (Algorithm [V.4](#)) is used to broadcast information contained in a payload to nearby connected devices whose identifiers are contained in the *endpointIds* list.

---

**Algorithm V.2** Discovering nearby device

---

```
1 public void startDiscovery (EndpointDiscoveryCallback callback) {
2     Nearby.getConnectionsClient ( activity )
3     .startDiscovery (NearbyConfig.SERVICE_ID, callback, NearbyConfig.getDiscoveryOptions ())
4     .addOnSuccessListener(new OnSuccessListener<Void>() {
5         @Override
6         public void onSuccess(Void aVoid) {
7             activity .handleDiscoveringStatus ( true );
8         }
9     })
10    .addOnFailureListener (new OnFailureListener () {
11        @Override
12        public void onFailure (@NonNull Exception e) {
13            activity .handleDiscoveringStatus ( false );
14        }
15    });
16 }
```

---

### 5.3 Needs calculation

As stated in the SocialMANET protocol [13], stations periodically broadcast *QUERY* announcement messages to signal their presence and these messages are used to calculate their respective needs in terms of missing publications. Since the discovery of nearby devices in our implementation is entrusted to the Nearby Connections API, the *QUERY* message is sent to devices as soon as a connection is established with them. However, as specified in the SocialMANET protocol, any new publication made by a device is immediately broadcast to all other nearby

---

**Algorithm V.3** Connecting to a nearby device

---

```
1 public void connect( final String endpointId,
2     ConnectionLifecycleCallback connectionLifecycleCallback ) {
3     String UserID = NearbyUtils.getUserNickName(activity .getApplicationContext ());
4     Nearby.getConnectionsClient ( activity )
5     .requestConnection (UserID, endpointId , connectionLifecycleCallback )
6     .addOnSuccessListener(new OnSuccessListener<Void>() {
7         @Override
8         public void onSuccess(Void aVoid) {
9             activity .handleRequestConnectionStatus ( endpointId , true );
10        }
11    })
12    .addOnFailureListener (new OnFailureListener () {
13        @Override
14        public void onFailure (@NonNull Exception e) {
15            activity .handleRequestConnectionStatus (endpointId , false );
16        }
17    });
18 }
```

---

---

**Algorithm V.4** Dispatching a payload in the network

---

```
1 public void sendPayload( final List<String> endpointIds , final Payload payload) {
2     Nearby.getConnectionsClient( activity )
3     .sendPayload(endpointIds , payload)
4     .addOnFailureListener( new OnFailureListener () {
5         @Override
6         public void onFailure ( @NonNull Exception e ) {
7             // retry
8             Nearby.getConnectionsClient ( activity ).sendPayload(endpointIds , payload);
9         }
10    });
11 }
```

---

connected devices. The construction of the *QUERY* message is done by the method *computeQuery* (Algorithm V.5).

---

**Algorithm V.5** Preparing the QUERY message

---

```
1 private Query computeQuery() {
2     Query query = new Query();
3     for (Topic topic : topicsManager.getSubscriptions ()) {
4         query.addEntry(new QueryEntry(topic.getId ());
5         for ( Publication pub: publicationManager.getPublications ( topic )) {
6             query.addPublication ( topic .getId (), pub.getPublicationId ());
7         }
8     }
9     return query;
10 }
```

---

When a device receives the *QUERY* message, it updates the information about the devices it is connected to, and then builds a list of publications to send and a list to inquire from the device that sent the *QUERY* message. These operations are respectively performed by the methods *computePubsToSend* and *computePubsToAsk* defined in the main activity Java class *MainActivity.java*.

## 5.4 Communication with the RRP

In our implementation, the RRP is a REST API deployed on a server on the Internet. It exposes two endpoints described as follows:

### 5.4.1 Endpoint 1 - <https://sm-rrp.herokuapp.com/rrp-api/v1/publications/get>

It is used to retrieve publications stored on the *RRP* that are missing from the station sending the request. Requests sent to this endpoint are made using the http *POST* method. Listing 1 shows the model and an example of the body of requests to be sent to this endpoint. This body is basically a JSON object with a unique property *entries*, which is a collection of query entries (*QueryEntry*). A query entry is made up by a topic ID (*topicId*) and a set of publication IDs concatenated into a single string and separated by "*.\_.*". The set of query entries represents the part of the *QUERY* message of the SocialMANET protocol delimited by brackets (see Fig. 1).

Listing 1: Endpoint 1 - Body

---

```

1  __Model__
2  Query{
3      entries : [QueryEntry]
4  }
5
6  QueryEntry {
7      topicId : String ,
8      publicationIds : String
9  }
10
11 __Example__
12 {
13     " entries ": [
14         {
15             " topicId ": " topic -id",
16             " publicationIds ": "pub-id-1_._.pub-id-2_._.pub-id-3"
17         }
18     ]
19 }

```

---

The content of the response to a request sent to this endpoint is a also JSON object. It has two properties named *publications* and *pubRequestIds*. The former is a set of publications matching the query received by the RRP as the request body. The later is a set of publication IDs that were present in the query received by the RRP and whose corresponding publication are unknown by it. The model and an example of the response to a request sent to this endpoint are given by listing 2. A publication here is made up by its ID (*publicationId*), the ID of the topic to which it belongs (*topicId*) the publication's date and validity in form of timestamps (*publicationDate*, *validity*), a *description* which is the content of the publication. Each publication can have an attachment file, although we haven't take it into account in our implementation for the sake of simplicity. The property *path* of the *Publication* object represent the path on a given station file system where this attachment is stored.

Listing 2: Endpoint 1 - Response

---

```

1  __Model__
2  RrpResponse{
3      publications : [ Publication ],
4      pubRequestIds: [ String ]
5  }
6
7  Publication {
8      publicationId : String ,
9      topicId : String ,
10     publishDate: Number,
11     validity : Number,
12     description : String ,
13     path: String
14 }
15
16 __Example__

```

```

17 {
18   "publications ": [
19     {
20       "publicationId ": "pub-id-1",
21       "topicId ": "topic-id",
22       "publishDate ": 1643459321851,
23       "validity ": 1646051522942,
24       "description ": "Sample description ",
25       "path ": null
26     }
27   ],
28   "pubRequestIds": ["pub-id-2", "pub-id-3"]
29 }

```

---

#### 5.4.2 Endpoint 2 - <https://sm-rrp.herokuapp.com/rrp-api/v1/publications/save>

This endpoint is used to send publications to the *RRP* which will then store them. It is useful when a station publishes an information while connected to the Internet, or after receiving a response to a request sent via the endpoint 1 for which the *'pubRequestIds'* field of the response is not empty. The format of the request's body and response is as in Listing 3.

Listing 3: Endpoint 2 - Body

```

1  __Model__
2  [ Publication ]
3  Publication {
4    publicationId : String ,
5    topicId : String ,
6    publishDate : Number,
7    validity : Number,
8    description : String ,
9    path : String
10 }

```

---

To ensure that the application is notified when a device is connected to a network, it has to subscribe to the *BroadcastReceiver*. This have been done in the main activity Java class *Main-Activity.java*. When this happens, the *handleConnectionStatusChanged* method is executed. This method is defined in the Java class *MainActivity.java* and depending on the status (on-line or off-line) of the device, it initiates a communication with the *RRP* as described in section 4.2.

## VI RESULTS

This section presents the results we obtained after the implementation of our SocialMANET revisited protocol. We implemented this protocol in a prototype application for information sharing in a scientific conference, *ConfInfo*. This application is made to run in Android smartphones. The source code is hosted on Bitbucket in a public repository at the following address: <https://bitbucket.org/martinxt/socialmanet/> (a README file which explains how to clone an run the code is available). The REST API that we proposed as the Remote Rendezvous Point

(RRP) is hosted on Heroku and the URIs of the different endpoints and their descriptions have been provided in section 5.4.

Figure 4 shows three screen shots of the *ConfInfo* application. The different essential parts that can be seen in figure 4(a) are labelled with numbers and described as follows:

1. Indicator which, when green, shows that the device is scanning the perimeter looking for nearby devices,
2. Indicator that when green, shows that the device is making announcements to show its presence,
3. Navigation bar to access the list of publications, the list of topics (current tab) and the list of nearby devices respectively,
4. List of topics, headed by topics to which the user has subscribed (Subscriptions).

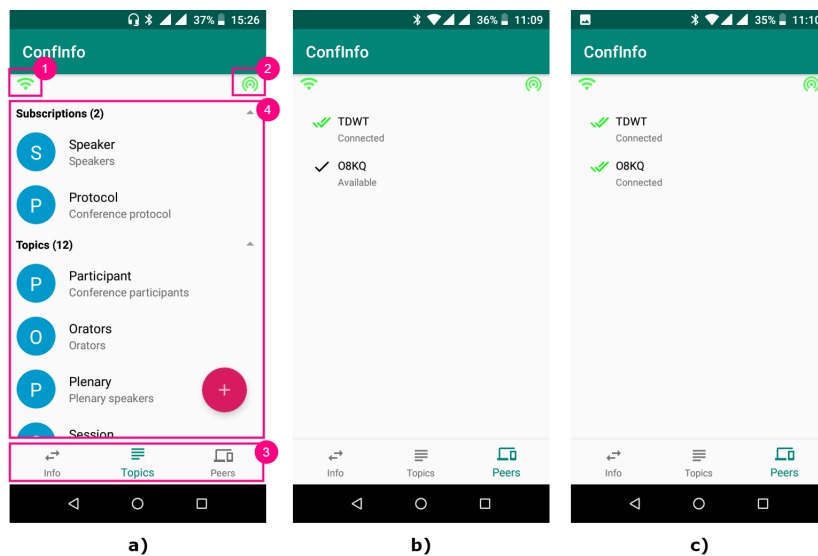


Figure 4: Application main parts and nearby devices

Figures 4(b) and 4(c) show a device having two devices available in its vicinity: *TDWT* and *O8KQ*. At 11:09, only *TDWT* is connected (Fig. 4(b)); one minute later, *O8KQ* is also connected (Fig. 4(c)).

Figure 5 demonstrate the capacity of the application *ConfInfo* of getting publications from nearby devices using Wi-fi Direct or Wi-fi P2P (Fig. 5 (a, b, c)) and from the RRP via Internet (Fig. 5 (d, e, f)). We have two devices that have both subscribed to the topics *Speaker* and *Protocol*. They can be distinguished by their battery level indicators (*WLU7* with a low battery level and *GBLT* with a high battery level).

Figure 5(a) shows the device *WLU7* publishing a message at 19:15. At 19:18, it gets near to the device *GBLT* and thanks to the communication protocol implemented in the application *ConfInfo*, *GBLT* will receive from *WLU7* the publication it made few minutes ago. Figure 5(b) shows *GBLT* receiving a notification indicating that a new publication has been received. Figures 5(d, e, f) show the same devices. They are not nearby each other. *GBLT* publishes a message at 19:20 while connected to Internet (Fig. 5(d) - notice the mobile data indicator). As a result, this publication is dispatched to the RRP and *WLU7* receives it few seconds later as it activates mobile data connection and gets connected to Internet (Fig. 5(e)).

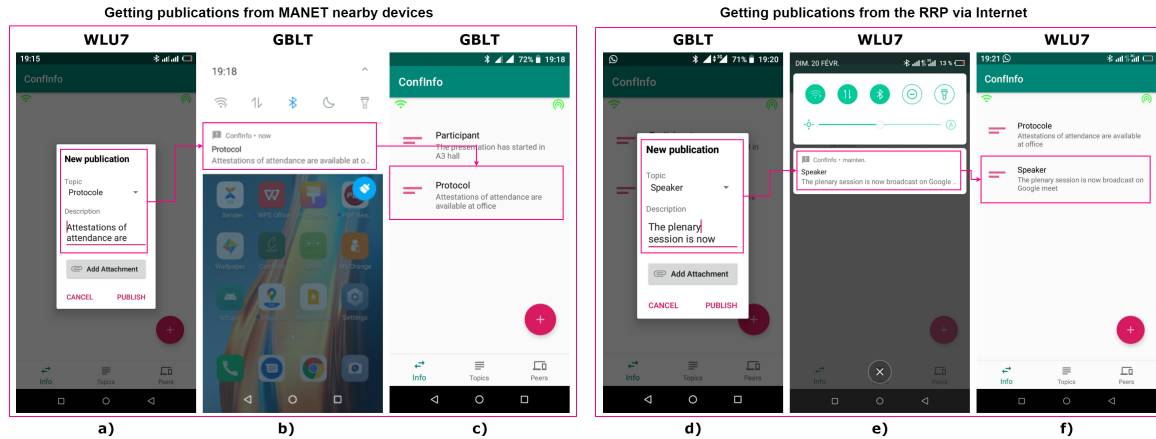


Figure 5: Publishing and dispatching publications

## VII CONCLUSION

In this paper, we have presented an opportunistic protocol for information dissemination in a hybrid ASN. Our protocol is based on the SocialMANET information dissemination protocol using a publish/subscribe communication model that we have extended to enable integration between ASNs and infrastructure networks. The implementation of our protocol in a prototype application for disseminating information at a scientific conference, ConfInfo, and the results obtained have enabled us to demonstrate its applicability. Nevertheless, it would be wise to continue with the validation of our solution by carrying out experiments in a real context in order to draw conclusions based on the results obtained in practice. In fact, the next African Conference on Research in Computer Science - *CARI'2022* - which will take place in October 2022 and will be hosted simultaneously on three sites (Tunis, Yaoundé, Dschang), could allow us to carry out experiments on the revised SocialMANET protocol proposed in this paper, as well as on our prototype ConfInfo application.

## ACKNOWLEDGEMENTS

The authors would like to express their sincere thanks to the anonymous reviewers for their insightful comments and valuable suggestions made on the earlier version of this paper presented at the Conférence de Recherche en Informatique CRI'2021.

## REFERENCES

### Publications

- [1] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications". In: *IEEE/ACM Trans. Netw.* 11.1 (Feb. 2003), pages 17–32. ISSN: 1063-6692.
- [2] K. U. Rahman Khan, R. U. Zaman, and A. V. G. Reddy. "Integrating Mobile Ad Hoc Networks and the Internet: challenges and a review of strategies". In: *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*. 2008, pages 536–543.

- [3] K. U. R. Khan, A. V. Reddy, and R. U. Zaman. “An Efficient Integrated Routing Protocol for Interconnecting Mobile Ad Hoc Networks and the Internet”. In: *Proceedings of the International Conference on Advances in Computing, Communication and Control*. ICAC3 '09. Mumbai, India: Association for Computing Machinery, 2009, pages 460–466. ISBN: 9781605583518.
- [4] I. Leontiadis, P. Costa, and C. Mascolo. “A Hybrid Approach for Content-Based Publish/Subscribe in Vehicular Networks”. In: 5.6 (Dec. 2009), pages 697–713. ISSN: 1574-1192.
- [5] T. Mishra, D. Garg, and M. M. Gore. “A Publish/Subscribe Communication Infrastructure for VANET Applications”. In: *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*. 2011, pages 442–446.
- [6] L. V. Hoang and H. Ogawa. “A platform for building ad hoc social networks based on Wi-Fi direct”. In: *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*. 2014, pages 626–629.
- [7] D. Zhang, D. Zhang, H. Xiong, C.-H. Hsu, and A. V. Vasilakos. “BASA: building mobile Ad-Hoc social networks on top of android”. In: *IEEE Network* 28.1 (2014), pages 4–9.
- [8] S. Gambhir, N. Aneja, and S. Mangla. “Need of ad-hoc social network based on users’ dynamic interests”. In: *2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI)*. 2015, pages 52–56.
- [9] X. Hu, T. H. S. Chu, V. C. M. Leung, E. C.-H. Ngai, P. Kruchten, and H. C. B. Chan. “A Survey on Mobile Social Networks: Applications, Platforms, System Architectures, and Future Research Directions”. In: *IEEE Communications Surveys Tutorials* 17.3 (2015), pages 1557–1581.
- [10] N. Aneja and S. Gambhir. “Middleware Architecture for Ad-hoc Social Network”. In: *Research Journal of Applied Sciences, Engineering and Technology* 13.9 (2016), pages 690–695.
- [11] J. A. Silva, J. Leitão, N. Preguiça, J. M. Lourenço, and H. Paulino. “Towards the Opportunistic Combination of Mobile Ad-Hoc Networks with Infrastructure Access”. In: *Proceedings of the 1st Workshop on Middleware for Edge Clouds and Cloudlets*. MECC '16. Trento, Italy: Association for Computing Machinery, 2016. ISBN: 9781450346689.
- [12] N. Aneja and S. Gambhir. “Profile-Based Ad Hoc Social Networking Using Wi-Fi Direct on the Top of Android”. In: *Mobile Information Systems* 2018 (Oct. 2018), pages 1–7. ISSN: 1875-905X.
- [13] M. T. Tchendji, M. X. Tchembé, and I. T. Necheu. “SocialMANET: A Publish/Subscribe Events Dissemination Protocol for Mobile Ad-Hoc Networks”. In: *Journal of Computer Science* 15.9 (Sept. 2019), pages 1237–1255.
- [14] M. Al mojamed and L. Dong. “Integrating Mobile Ad Hoc Networks with the Internet Based on OLSR”. In: 2020 (Jan. 2020). ISSN: 1530-8669.
- [15] M. T. Tchendji, M. X. Tchembé, and A. L. M. Kakeu. “An ad-hoc social network generation approach”. In: *CoRR* abs/2008.04618 (2020). arXiv: 2008.04618.
- [16] N. Aneja and S. Gambhir. “Recent Advances in Ad-Hoc Social Networking: Key Techniques and Future Research Directions”. In: *Wireless Pers Commun* 117 (2021), pages 1735–1753.
- [17] R. Xing, Z. Su, Q. Xu, and A. Benslimane. “Truck Platooning Aided Secure Publish/Subscribe System Based on Smart Contract in Autonomous Vehicular Networks”. In: *IEEE Transactions on Vehicular Technology* 70.1 (2021), pages 782–794.