Conception d'un support de communication opportuniste pour les services pervasifs

Ali Makke

THÈSE / UNIVERSITÉ DE BRETAGNE SUD
UFR Sciences et Sciences de l'Ingénieur
sous le sceau de l'Université Européenne de Bretagne

Pour obtenir le grade de :
DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE SUD
Mention : Informatique
École Doctorale SICMA

présentée par

Ali MAKKE

IRISA Institut de Recherche en Informatique et Systèmes Aléatoires

Pervasive Service Provisioning in Intermittently Connected Hybrid Networks

Conception d'un support de communication opportuniste pour les services pervasifs

Thèse soutenue le 03 Mars 2015,
devant la commission d'examen composée de :

M. Jean-Marie BONNIN
Professeur, Télécom Bretagne / Président

M. Joan BORRELL
Professor titular, Universitat Autònoma de Barcelona / Rapporteur

M. Stéphane FRÉNOT
Professeur des Universités, INSA de Lyon / Rapporteur

M. Yves MAHÉO
Maître de Conferences HDR, Université de Bretagne-Sud / Directeur de thèse

M. Nicolas LE SOMMER
Maître de Conferences, Université de Bretagne-Sud / Encadrant de thèse

Contents

List of Figures

List of Tables

Part I

Context of the Thesis

1 Background and Motivations

Today, the Internet is becoming more and more pervasive. Smarter devices are gradually conquering the environment around us and completely changing the way we communicate. All these new devices are designed to be connected to the Internet through different interfaces, such as Wi-Fi and Bluetooth, in the aim of being accessible at anytime and from anywhere. Moreover, these devices are typically capable of interacting with each other, thus creating an interactive smart space that aims at improving our interactions our physical environment. This paradigm is known as "pervasive computing".

This evolution has hugely affected the people's behavior, as they constantly want to share different kinds of information (e.g., movies, pictures and music), to benefit from the different services present around them (e.g., checking weather and traffic information) or even to control the different devices present in their environment (e.g., lights, TV or even door locks). Users want to perform all these tasks using the smartphone in their hands as they consider it like the Swiss-Army knife for technology. In fact, people nowadays are more connected to their smartphones as we can somehow say that a smartphone is becoming a "natural extension" of a person. As a consequence, the Internet should be able to support this evolution.

1.1 Pervasive Computing

The notion "pervasive computing" was initially envisaged by Mark Weiser [130, 131, 132]. Pervasive computing aims at enhancing computer usage by making many computers available throughout the physical environment, but making them invisible to the user. The essence of this vision was the dream of having an environment where traditional networking technologies will complement new advanced computing and wireless communication capabilities, while being in line with the human users needs.

The research path towards making pervasive computing a complete reality is still long and winding. Currently, pervasive computing has started to gradually integrate in our lives through everyday objects, aiming that all the devices embedded in the environment should be controlled using smartphones and tablets. For example, a lot of work have been already made on home automation or domotic. Thanks to home automation, we can control a large number of devices by relying only on our smartphone such as TVs, radios, heating, lighting, security, etc (Figure 1.1).

Figure 1.1: Example of home automation (domotic)

For pervasive computing to accomplish its goal of seamlessly integrating in the physical environment, the Internet itself should evolve along with the emerging ubiquitous environment and become an Internet of Things. Indeed, the Internet should be prepared to absorb the huge amount of devices to be connected to it. Therefore, pervasive computing environments will become saturated with computing and communication capabilities. Users, on the other hand, should be able to benefit from their desired services anytime and everywhere. They should be able to discover the resources and services available in the surrounding environment, and beyond that they should be able to connect to distant services and resources located in remote areas from their physical position.

Let us consider a city center scenario to point out a sample of the case studies we target in our work. Bob is visiting France on vacation. His objective is to visit touristic attractions in a couple of days, so he spends most of his time outside. He uses his smartphone to search for the closest historical monument to start his tour with, how to arrive there and check some attraction recommendations. During launch time, Bob searches restaurants in his vicinity. Since he follows a strict diet, Bob wishes to find a restaurant that provides food that meets his dietary regulations and offers complementary Wi-Fi access. Bob typically plans the remainder of his day while waiting for his meal to be served by exploring services that offer tourist recommendations that include photo snapshots, video trailers and visitor feedback. Unintentionally, the second day of Bob's visit to this city happens to coincide with a traditional local festival. In this festival, different spectacles take place all over the city center. While some street performers specialize in acting or playing traditional music, others tend to invite people from the audience to participate in the shows. By searching on his smartphone, Bob is able to find some shows that he can participate in, search for vendors that engage in the festival to sell their handmade

traditional products, and even book a ticket to have a horse-drawn carriage ride through the streets of the old city as a part of the event.

In reality, the scenario we introduced here is partly achievable. Indeed, the part related to searching for touristic attractions and finding some nearby restaurants is fairly common nowadays, however, the part related to finding the different performers and sellers participating in a festival or reserving a ticket for a local activity is not quite common or easily available. In fact, this part of the scenario should be possible as plenty of communication means that are needed to make it a reality are already available such as 3G, Wi-Fi, Bluetooth, etc.

1.2 Problem Statement

The pervasive computing paradigm has led to the creation of pervasive environments that are subjected to several challenges. Accessing the services present in these environments is one of the major challenges we are dealing with in this thesis. We refer to this process with the term "service provisioning". Unlike provisioning in other domains, service provisioning in our work is the process of discovering, selecting and invoking software services from a protocol point of view (i.e., introducing protocols that allow the discovery of software services by potential clients in the network and invoking them when needed). The different parts consisting the provisioning process will be defined in details in Chapter 2.

The mainstream approach for service provisioning is to rely on the infrastructure supported by service operators. This can be applied on most of the cases of the city center scenario we introduced before. For instance, the most straightforward way, till now, for a user to connect to services offered by service providers is by relying on the 3G or 4G (LTE) technologies. Truly, a great effort has been put on the infrastructure dedicated to service provisioning in such scenarios. Such an infrastructure, based on 3G/4G networks, include application servers and even positioning equipments. For instance, the positioning of both a client and a service provider might be shared with the servers of the service operators. In turn, these servers will compare these positions and inform the client about the set of closest services to it. Indeed, great research and industrial efforts have been concentrated to propose solutions for service provisioning along this line.

Relying exclusively on wired and wireless infrastructures (e.g., cellular networks, 3G/4G networks, or one-hop WLANs) to support the seamless integration of pervasive computing in the physical environment seems not to be suited, and not even feasible. Moreover, such an approach mainly imposes a large number of constraints on the clients (mobile users) either in terms of bandwidth or radio coverage. Furthermore, relying on such networks obligates clients to be always dependent on service operators through mandatory subscriptions to benefit from services. For example, a user subscribing to a service to check the movements of his child in the city center. This dependence on service operators rises might several problems, such as service interruptions due to malfunctioning servers, recording of personal data on operators' servers, expensive subscriptions etc.

In our work we investigate alternative solutions for accessing remote services from a network point of view. We focus on using the capability of the devices to directly communicate together. Thus, a user (the client to benefit from the provided services) is

not obliged to have a contract with a Global System of Mobile Communications (GSM) operator company or getting a 3G/4G connection. Instead, this user could benefit from the communication provided by his device interfaces to discover and interact with the provided services in his environment. In addition, several scenarios exist where local and direct communications do not act as alternatives to 3G/4G communications but as more suitable approaches to be relied on than other types of communication. For instance, obtaining an Internet access is not always possible for the market place seller presented in our scenario. Instead, the seller could rely on a personal computer, also known as an infostation[1], to provide and advertise his services directly by communicating with the smartphones of the nomadic clients roaming the streets.

The capabilities of such devices to communicate together, and the opportunity of "unlimited" pair-wise contacts among them, could be used to reach a suitable approach of accessing resources in such environments, often referred to as "opportunistic networking". In some cases, exploiting the opportunistic pair-wise contacts between the devices forming the network might be cost effective in comparison to the potentially high costs of infrastructure. From a provider (i.e, infrastructure) point of view, it is unclear if deploying some small infostations is noticeably cheaper than relying on the actual existing infrastructure of the service operators. Indeed, it is difficult to have a cost model as it depends on the scenario and the targeted environment (e.g., urban or rural areas, city centers, etc.). As for the client, he will often consider that it is cheaper to rely on direct communication using Wi-Fi interfaces than relying on the 3G subscription to exchange information with a somehow close device.

However, such local communication mechanisms are rarely used in practice, especially because they introduce frequent connectivity disruptions, which break end-to-end paths between the different devices in the network. These connectivity disruptions mainly result from the free movement of people, the short communication range of wireless interfaces, the radio interferences and the volatility of the devices—which are frequently switched off in order to limit their power consumption, leading to challenging scenarios of frequent *disconnections* and significant *delays*. For that, in order to exchange messages in an opportunistic network, nodes rely on the "store, carry and forward" concept inherited from delay/disruption-tolerant networks (DTNs) where they dynamically build the routes traversed by the message from the sender toward the destination(s). Any possible node can opportunistically be used as intermediate relay, if it is likely to bring the message closer to the final destination(s)[2].

From a broad perspective our work aims at finding an answer to the question: *How to support pervasive service provisioning relying on the opportunistic networking approach?*

In fact, opportunistic networking is considered a recent domain and has not been, for the moment, largely used for the provisioning of pervasive services. As introduced by Conti et al. [20], providing services based on opportunistic networking is itself a wide research topic that has been named "Opportunistic Computing". In practice, efforts have been focused on performing efficient communication between devices in opportunistic

[1] An infostation is the device a service provider relies on to provide his services, such as a fixed device equipped with various wired and wireless interfaces.

[2] In this document we interchangeably use the terms forwarding and routing as the forwarding process performed by a node on any message is the result of a routing decision performed by the protocol implemented in the node.

networks. For that, we intend to benefit from the concepts that emerged in this opportunistic networking community and exploit them in the context of pervasive service provisioning.

With the aim of extending the concept of opportunistic networking, environments are modeled as a collection of resources that are exposed as application services. Following a Service-Oriented Architecture approach, this allows users to exploit resources offered by the static or the mobile devices that appear in their surrounding environment. However, an inversely proportional relationship exists between the ease of mobility and the capabilities of the device. For instance, the available memory, battery and CPU power are often scarce and limited in mobile devices, while fixed devices such as desktops (equipped with wireless interfaces) have the capability to be equipped with more powerful (but harder to move) hardware. As a consequence, a special kind of networks can be introduced, known as Intermittently Connected Hybrid Networks (ICHNs). The most suited computing paradigm for ICHNs is an instance of the opportunistic computing that takes advantage of the fact that some devices are stationary and connected together (e.g., access points, DSL gateways), and the most of the services will be provided by these fixed devices. In this thesis, we focus our attention to ICHNs as they reflect a widely deployed realistic scenario, such as city centers, large companies, university campuses etc.

How to exploit the specific characteristics of ICHNs to optimize the pervasive service provisioning in an opportunistic way?

We focus on the various problems of opportunistic computing in ICHNs, such as connectivity disruptions, no end-to-end paths and absence of access continuity due to user mobility. The specific characteristics of ICHNs include having several fixed devices (infostations) and mobile devices (clients). This might result multiple connections between a user and fixed infostations. We exploit this behavior to ensure access continuity in ICHNs. In general, in 3G/4G and mesh networks, access continuity is ensured by relying on handover mechanism. For that, we investigate how such a mechanism can help at improving service provisioning in ICHNs. Furthermore, we aim at performing a light and fast dissemination mechanism of pervasive services over all potential clients in the network. As from a client point of view, we aim at having an efficient interaction with the provided services despite of the disconnected nature of the network. Finally, all designed protocols should be able to preserve both scalability and robustness.

1.3 Contribution and Outline of the Thesis

The remainder of the thesis is organized as follows. In chapter 2, we trace the evolution of wireless networks, discussing the advantages and challenges associated with each type of networks and specifying the network kind we consider in our work. We also introduce the issues of opportunistic computing and the service-oriented architecture in the context of our work.

In chapter 3, we survey works related to communication in intermittently connected networks, focusing on routing protocols targeting delay/disruption tolerant networks and opportunistic networks. In chapter 4, we give an overview of the research works dealing with opportunistic computing and address the solutions designed for service provisioning in different kinds of networks. We also highlight the various handover

mechanisms for service provisioning, concluding with a discussion about their adequateness for our case studies.

In chapter 5, we discuss in details our protocol dedicate for service discovery in Intermittently Connected Hybrid Networks. In chapter 6, we present the second protocol that tackles the service invocation challenge in such networks, describing the mechanisms and heuristics adopted for this solution. In chapter 7, we introduce the handover mechanism specifically designed to ensure access continuity for opportunistic computing, presenting the key design choices. In chapter 8, we introduce the representation of the middleware platform for service provisioning in ICHN we propose followed by the description of the service management application programming interface (API). In addition, each model of our framework is validated through a series of evaluations and comparisons with existing approaches. In chapter 9, we draw our conclusions about the main contributions of the thesis, and discuss some research perspectives and open questions leveraged by our work.

2

Toward Pervasive Opportunistic Computing

Intermittent connectivity is one of the major constraints that should be dealt with when targeting opportunistic computing in pervasive environments. In the following, we introduce the evolution of the wireless networks over the past few decades, specifying their limitations for handling the view of the future Internet we depicted in Chapter 1. We also define more precisely the network that we consider in our work, its characteristics, and our assumptions. Then, we discuss the constraints that raise from performing the service provisioning process in challenged networks. We conclude by justifying our choice of the network type focusing on the issues that should be tackled and what are the expected benefits for end-users.

2.1 Intermittent Connectivity

2.1.1 From Single-hop to Mobile Ad Hoc Networks

2.1.1.1 Single-hop Networks

Over the past decades, wired access networks were broadly extended by Wi-Fi hot-spots to integrate mobile devices. These hot-spots, also known as wireless access points (APs), offer a great way for mobile devices to get network services (see Figure 2.1). This concept has become common in public places such as coffeehouses, libraries and airports, where the constraints on the people's movement has been slightly reduced. In general, such APs are mainly deployed in order to provide nomadic people with an Internet access. Nevertheless, the constraints on mobility are still applied as a person must be present in the radio range of an AP to access the Internet. The radio range of APs is most of the time limited to 100 m.

In turn, mobile operator companies also moved toward supplying their clients with network service access resorting to their cellular infrastructures. These networks follow the same concept deployed in the WLAN, where devices remain connected to the network as long as they are in the coverage range of the base transceiver station (BTS), as shown in Figure 2.2. Less constraints are imposed on the mobility as several BTS towers are positioned so that their coverage area stretches over large cities. Thus, people are able to move freely while being seamlessly connected to the Internet. Despite such an advantage, these networks provide their clients with limited bandwidth, for example, UMTS bandwidth is less than 384 kb/s in most situations, sometimes due to the huge num-

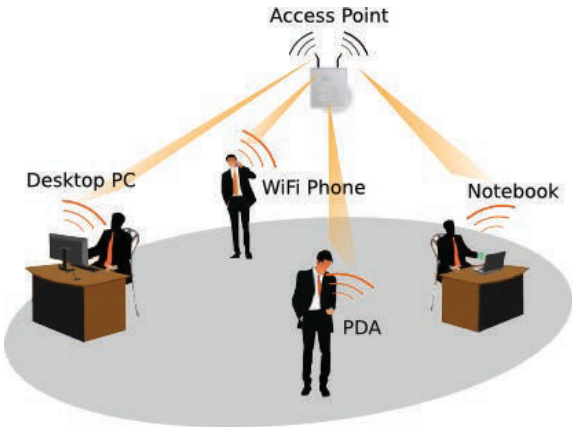


Figure 2.1: Example of a set of devices in the range of an access point (WLAN)

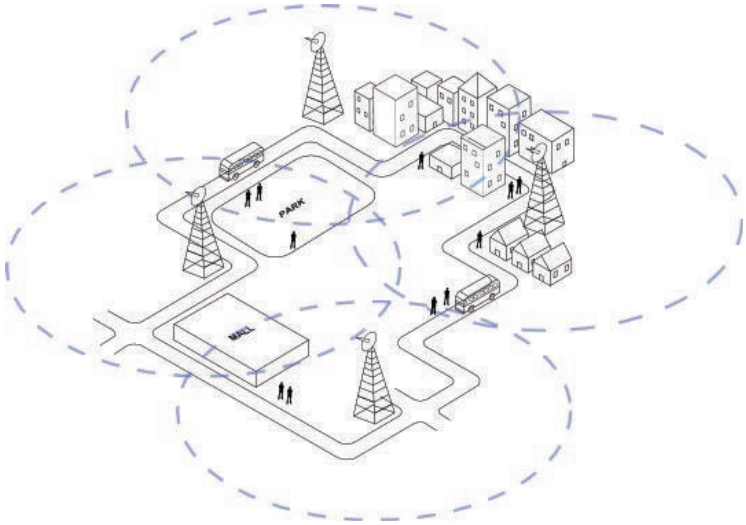


Figure 2.2: Users in the range of base transceiver stations (GSM network)

ber of devices joining the network. To overcome this limitation, 3G and 4G technologies have been recently introduced to provide people with better bandwidth. For instance, LTE Advanced is data-oriented, it aims to improve data speeds and to bring more network capacity for more data per user. Although such technologies provide users with higher bandwidth, this is done by using a mix of macro cells and a lot of small cells. The potentially high cost of setting up the necessary infrastructure and the time needed to set them everywhere impose high costs on the end users that want to benefit from such technologies.

Triggered by the emergence of low-cost wireless network interfaces in the consumer electronics market, new communication concepts gained interest in a relatively short period of time. One of the concepts that have been developed in this era is the mobile ad hoc networks (MANETs). These networks are described below.

2.1.1.2 Mobile Ad Hoc Networks (MANETs)

Mobile ad hoc networks (MANETs) are networks formed spontaneously by an autonomous set of devices that are connected via wireless links without relying on any pre-configured infrastructure or centralized administration. In other words, these mobile nodes equipped with one or more wireless interfaces (typically 802.11 in ad hoc mode) can freely and dynamically self-organize into arbitrary and temporary network topologies, allowing themselves to seamlessly communicate with each other in areas with no pre-existing communication infrastructure. As depicted in Figure 2.3, in a MANET, each node communicates directly with any other node within its transmission range, while communication beyond this range is established by employing intermediate nodes to set up a path in a hop-by-hop manner.

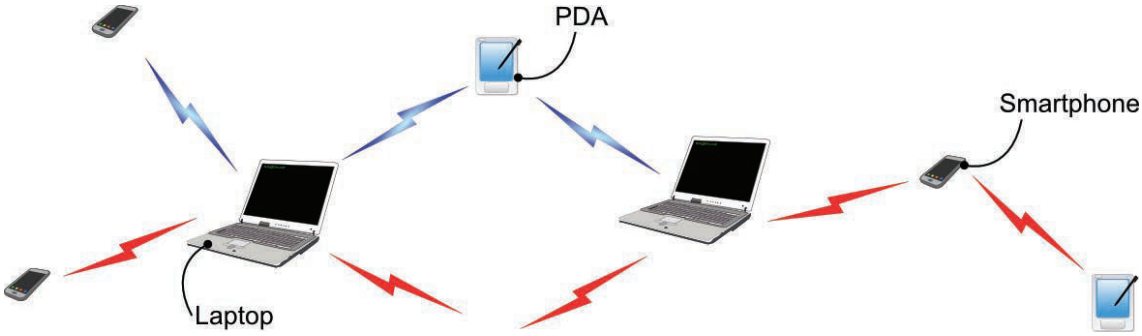


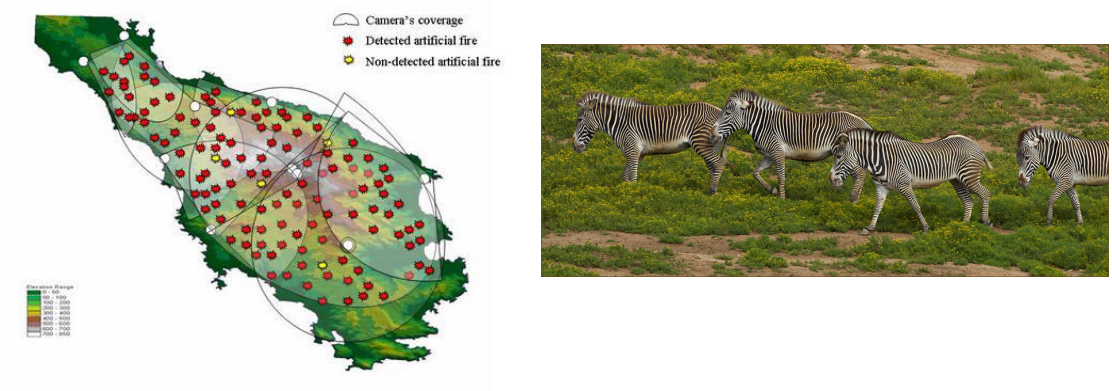
Figure 2.3: Example of a Mobile Ad Hoc Network with a multi-hop path between devices

Scenarios and Applications for MANETs

Among the common MANET usages is communication between a group of persons in proximity such as the case of soldiers in a battlefield or researchers attending a seminar. Vehicular communication represents one of the new challenging applications for MANETs, vehicle collision warning is one of the very promising potential applications.

Other forms of MANETs are Wireless sensor networks (WSNs) and wireless sensor and actuator networks (WSANs). The principle idea behind these networks is to equip miniaturized and battery-enabled sensor devices and actuators with memory as well as processing and wireless transmission capabilities. Thus, enabling these sensors and actuators to communicate with a data center when deployed in areas that are otherwise hard to reach. Fire detection in forests (Figure 2.4a) and data collection from wildlife (Figure 2.4b) are some examples of such networks.

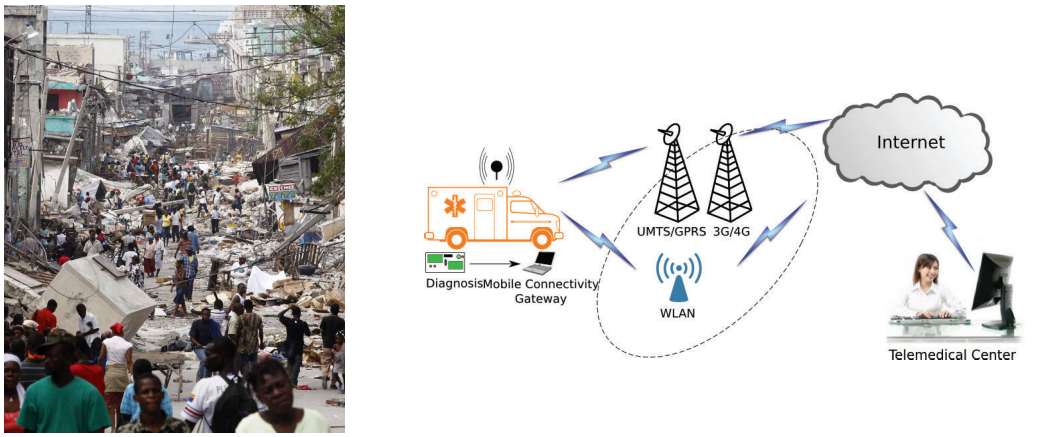
For example, distributing fire detection sensors in the forests (Figure 2.4a) or collecting data from wildlife (Figure 2.4b). Other applications fall in the category of communications in natural disasters relief situations where no communication infrastructures might be present and communications to organize the rescue teams are needed (Figure 2.5a).



(a) Fire detection in forests (b) Wildlife Data Collection

Figure 2.4: Different applications of MANETs

Beside the pure mobile structure of MANETs other hybrid forms of networks may exist, where MANETs can be also utilized as an extension to the infrastructure-based networks to overcome the coverage range limitations of the APs, thus extending both the range of the infrastructure and the service coverage to non-infrastructure areas. A sample application of hybrid networks is the vehicle-to-infrastructure communications that can be used in vehicular emergency applications, where being life-critical, these applications require connectivity throughout all the possible communication technologies (Figure 2.5b). Various projects based on MANETs already exist, such as Serval project [34], Open Garden [7], Roofnet [3], etc.



(a) Natural disasters relief (b) Vehicle-to-Infrastructure communications

Figure 2.5: Different applications of MANETs

Routing Protocols for MANETs

In the last decade, researchers have proposed a wide range of routing protocols for MANETs. The main goals of these protocols are: maximizing throughput while minimizing packet loss, control overhead and energy usage. In the following section we present the different approaches dedicated to routing in MANETs.

In Source-initiated routing protocols, the route is created only when a source requests a route to a destination node. It is created through a route discovery procedure which involves flooding the network with route request packets. One of the most known source-initiated protocols is the **Ad hoc On-Demand Distance Vector (AODV)** routing protocol [101]. Table driven protocols, on the other hand, always maintains up-to-date information of routes from each node to every other node in the network. One example of table driven protocols is the **Optimized Link State Routing (OLSR)** [18]. The hybrid routing schemes combine elements of on-demand and table-driven routing protocols. The general idea is that the area where the connections change relatively slowly are more amenable to table driven routing while areas with high mobility are more appropriate for source initiated approaches. By appropriately combining these two approaches the system can achieve a higher overall performance. The **Zone Routing Protocol (ZRP)** [40] is an example of hybrid protocols.

Location-Aware protocols assume that individual nodes are aware of the locations of all the nodes within the network. The best and easiest technique is the use of the Global Positioning System (GPS) to determine exact coordinates of these nodes in any geographical location. This location information is then utilized by the routing protocol to determine the routes, such as in **Location-Aided Routing (LAR)** [59] protocol. Scenarios such as soldiers in the battlefield or VANETs are mainly interested by such protocols.

Multipath routing protocols create multiple routes from source to destination instead of the conventional single route discovered by other protocols. Multiple paths are generated on-demand or using a pro-active approach. The extension of AODV called **Ad hoc On-demand Multiple Distance Vector (AOMDV)** routing protocol **[82]** is suited for multicast communication (i.e., for simultaneous transmission of data from one sender to multiple receivers). Several widely used applications require multicasting at least at the logical level. Examples include audio-video teleconferencing, real-time video streaming and the maintenance of distributed databases. Furthermore, geographical multicast routing is a variant of multicast where the goal is to route the packets coming from a source to destinations located within a specific geographic region. One solution is proposed in [74], which based **a geocasting protocol for mobile ad hoc networks (GeoGRID)** on the unicasting routing protocol GRID [75]. Naturally, for geocast to work, the nodes need to rely on localization techniques (such as GPS). Examples of applications that rely on such protocols are geographic messaging and advertising.

Discussion A large number of routing approaches have been proposed for MANETs. Nevertheless, all these approaches are built on the assumption that end-to-end paths are a given, and there exists at least one path between any source and destination in the network at all times. In practice, it is not always possible to maintain an end-to-end path between a source and destination due to nodes mobility and volatility.

MANETs have changed the classical centralized wireless network topology into a whole new domain with many potential applications. However, in spite of the research efforts, the MANET technology has only a marginal role in the wireless networking field and it has not been widely deployed for civilian usage until today. This can be justified by the lack of real applications in MANETs. One of the reasons of this lack is the expectation that mobile nodes are responsible of providing the applications and services. However, the lack on memory and computational power in mobile devices limits such a role. Another reason is the assumption that a client can instantaneously reach any provider in the network and use all the provided services.

2.1.2 Intermittently Connected Networks (ICNs)

In most real world mobile environments, intermittent connections are more certainly to occur when no restrictions are imposed on the users' behaviors. Yet, one of the key assumptions of MANETs is that despite of the dynamic network topology, end-to-end paths are always available between any two devices wishing to communicate. MANETs are implicitly assumed to be dense enough and formed of nodes with low mobility so they can be viewed as a fully connected graph.

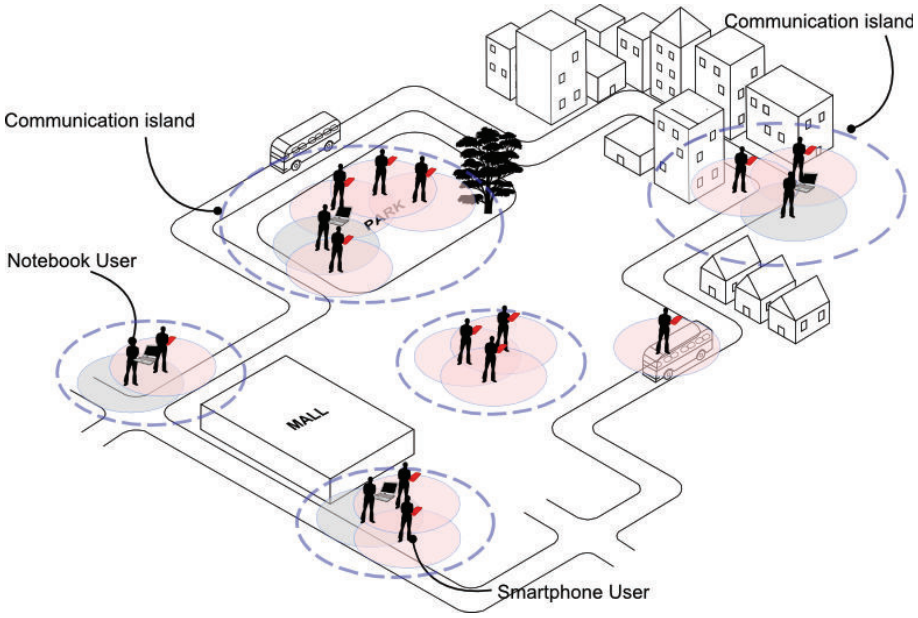


Figure 2.6: Illustration of Intermittently Connected Mobile Ad Hoc Networks (IC-MANETs)

In practice, this assumption reveals itself unrealistic and untenable in cases where the motion of the devices can result in disruptions of connections due to several constraints, such as: intervening objects, limited communication range of wireless interfaces, radio interference, volatility of devices that are frequently switched off in order to limit the

power consumption, and so on. For example, a military mobile ad hoc network may become intermittently connected when mobile nodes (e.g., soldiers, tanks) move out of the transmission range of each other. In addition, similar intermittent disconnectivity can be encountered by VANETs due to high vehicle velocities (compared to the transmission range). Consequently, it is most likely to end up with a network formed of several distinct islands of nodes rather than a fully connected graph, thus entail the creation of a type of network known as Intermittently Connected Mobile Ad Hoc Network (ICMANET).

2.1.2.1 Intermittently Connected Mobile Ad Hoc Networks (ICMANETs)

An ICMANET is a network purely formed of mobile devices, that are highly mobile and/or have a quite sparse distribution in the network space, as shown in Figure 2.6. Such networks are characterized by their lack of connectivity and are eventually formed without any assumption about the existence of an end-to-end path between any pair of nodes, for it is perfectly possible that two nodes may never be part of the same connected portion of the network. A practical example of such networks is a pocket switched network [15] that is formed by human carried mobile devices based on their Bluetooth or IEEE 802.11 interfaces with short transmission range. Due to this lack of connectivity, ICMANETs share a common characteristic with Delay Tolerant Networks, where the incurred delays can be very large and unpredictable.

As a consequence, communication in these intermittently connected networks cannot be achieved thanks to traditional routing protocols designed for wired networks or thanks to dynamic routing protocols such as AODV (Ad hoc On Demand Distance Vector) or OLSR (Optimized Link State Routing Protocol), since under such protocols, if the source node cannot discover the corresponding end-to-end path, the required data session cannot be facilitated. Moreover, when packets arrive at intermediate nodes and no contemporaneous path to their destination can be found, these packets are simply dropped.

The mobility of devices is considered as one of the main challenges for conventional MANETs, it needs to be handled properly to enable seamless communication between the devices. However, the mobility in ICNs is considered as an assistant to overcome the intermittent connectivity in such networks. Indeed, it is recognized as a critical component for data communications between the nodes that may never be part of the same connected portion of the network. Devices communicate directly when they are in range of one another, and intermediate nodes can be used to relay a message following the "store, carry and forward" principle. Routes are therefore computed dynamically at each hop while messages are forwarded toward their destination(s). Each relay node receiving a message is thus expected to transmit a copy of the message to one or several of its neighbors. When no forwarding opportunity exists (e.g., no other nodes are in the transmission range, or the neighbors are evaluated as not suitable for that communication) the node stores the message and waits for future opportunities with other devices to forward the message. Thanks to this principle, a message can be delivered even if the source and the destination are not present simultaneously in the network, or if they are not in the same network island at emission time. This principle helps overcoming the disconnectivity problem but rises several other issues, such as limiting the overhead produced from message dissemination in the network, increasing the satisfaction ratio or the successful

delivery percentage of messages between clients and providers and reducing the amount of time needed to deliver a specific message from the source to the destination. All such issues are significant and should be tackled when designing any protocol targeting these networks.

The fact that ICMANETs are purely formed of mobile devices imposes limitations on the provided service and applications. Similar to MANETs, the devices roaming the ICMANETs are expected to provide the applications and services themselves. This is reflected as a drawback of the network as the majority of the devices have scarce resources and limited computational power.

2.1.2.2 Intermittently Connected Hybrid Networks (ICHNs)

Another specific class of Intermittently Connected Networks (ICNs) is what we call Intermittently Connected Hybrid Networks (ICHNs). An ICHN is an extension of an ICMANET. It is an infrastructure-based network with opportunistic extensions: it includes some fixed infostations (potentially connected together with some fixed infrastructure, typically the Internet) and a set of mobile devices, viewed as an ICMANET that is exploited with opportunistic networking techniques, as depicted in Figure 2.7.

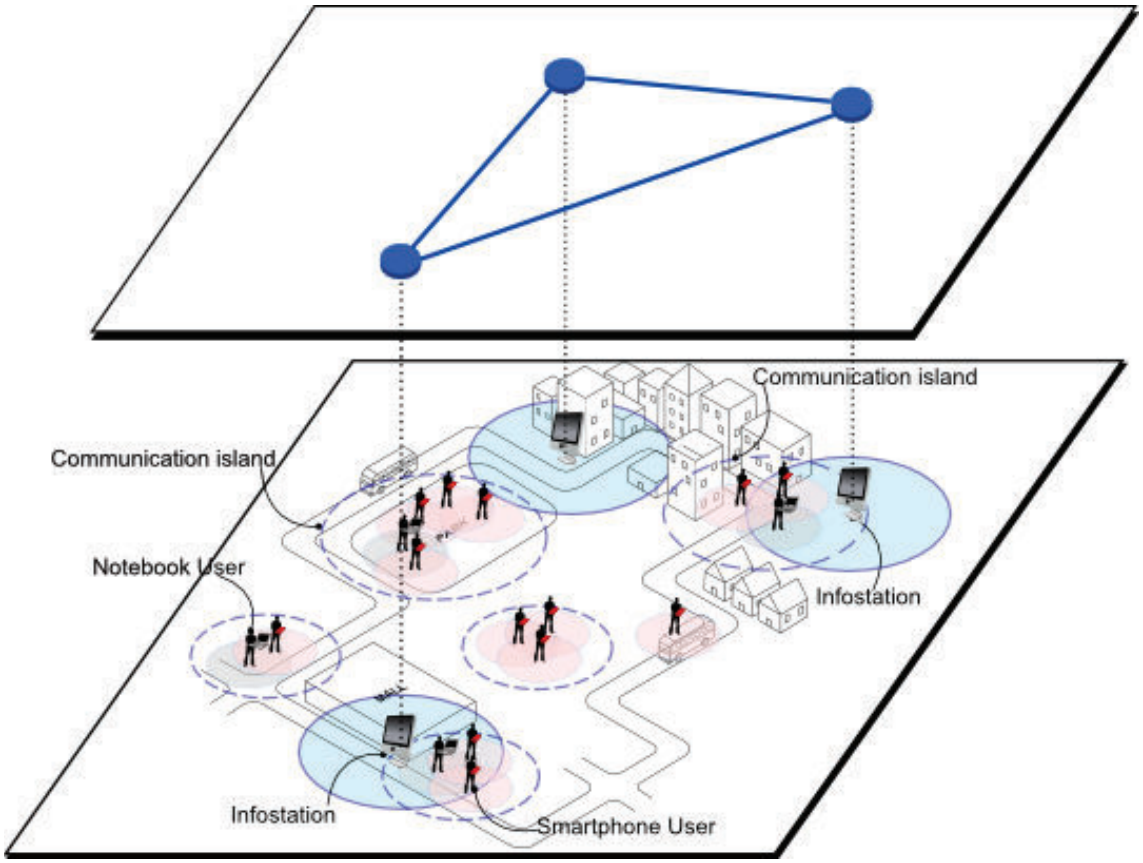


Figure 2.7: Example of Intermittently Connected Hybrid Networks (ICHNs)

Such network type can be present in various scenarios in our daily life. For example, taking the case of a city center with various kinds of shops (restaurants, hotels, agencies) and pedestrians roaming the streets of the city. The fixed infostations in this scenario can be represented by pcs placed within the various shops in the city, some of these pcs may be directly connected to the Internet, others may be connected to other branches of the same business through a private network. These infostations are used to provide services to the clients roaming the city streets. Pedestrians, in turn, equipped with their smartphones represent the ICMANET part of the network, where they can move freely at walking speeds (0.5 to 2 mps) and benefit from the services provided by the infostations around them. Such a scenario can be applied in various real world environments.

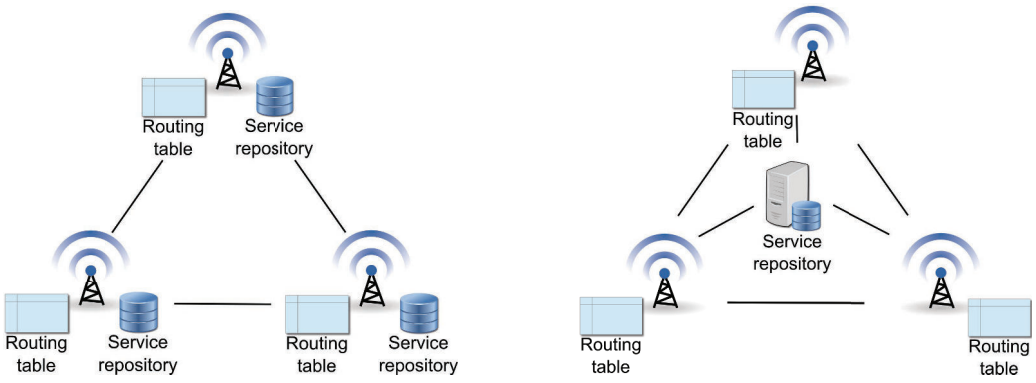
ICHNs can be seen as an opportunity for service providers, such as local authorities, to provide nomadic people with new ubiquitous services, without resorting to expensive infrastructure such as cellular networks. In fact, ICHNs share the same view of future Internet with the concept of opportunistic computing that allows users to exploit the resources offered by the static or the mobile devices available in the surrounding.

Since ICHNs are considered as extensions of ICMANETs, they inherit their characteristics and challenges in terms of the lack of connectivity and absence of end-to-end paths between the nodes forming the network. Thus similarly, the "store, carry and forward" principle should be utilized for transmitting messages among the various intermediate nodes in the network till reaching the final destination of each message.

However, ICHNs are distinguished from ICMANETs with the presence of a fixed part in the network. The presence of infostations can be viewed as an advantage in terms of message delivery. Indeed, having fixed and reliable infostations with interconnections among different subsets of these infostations can be exploited to overcome various challenges in routing and guiding the flow of messages in the network to obtain a better and faster message delivery process.

From our point of view, an ICHN represents one of the most interesting and realistic type of network for the future Internet. In contrast, the pure composition of ICMANETs of mobile devices solely poses a lot of constraints on the number of services that can be offered by such type of networks to the mobile users. Indeed, these devices are characterized with scarce resources, limited computational powers and short communication range of their wireless interfaces. Besides, the mobility and the volatility of these devices result low density networks with some frequent and unpredictable disruptions in the communication links and without any assumption about the existence of an end-to-end path between any pair of devices. Thus, due to these constraints it can be legitimately considered that only few services will be provided directly by the mobile devices carried by people.

Depending of the scenario of the ICHN network not all the infostations present in the network are imperatively connected together. Indeed, depending on the service providers, some infostations might be connected together (directly through a private network, or using the Internet), others might just be isolated and unaware of the presence of any other infostation in the network.



(a) Architecture example 1 (b) Architecture example 2

Figure 2.8: Infrastructures of infostations

Infostations are assumed to be continuously running, never out of order and powerful enough to perform computational tasks. Moreover, they can present various topologies (mesh, bus, etc.), and service repositories can be organized in a centralized or distributed manner. For example, the infostations can provide the services themselves and can act as service repositories (see Figure 2.8a), or can simply act as gateways for other providers that register their services with a centralized repository (see Figure 2.8b). No constraints are made on the physical positions of these infostations, as this is related to the service providers willing to provide services. As a consequence, we assume that the density of infostations is low with sparse distribution in the physical environment.

Since service providers might be unrelated local companies, different infostations might not be aware of each other especially if one or more of them are not connected to the Internet. Thus, two extreme scenarios might exist:

1. The case where all the infostations in the network are connected to each other (or the Internet), thus the resulting infostations might act as proxies to each other.
2. The case where none of the infostations are connected to the others, thus resulting in the presence of isolated infostations in the network.

In our work, we consider a generalization covering these two cases (see Figure 2.9). We take into account the presence of both types of communications in the network. Some infostations might be connected to the Internet, others privately connected together (e.g., infostations of several branches of the same company placed in different parts of the city and providing a specific set of services) and a set of isolated infostations. Indeed, our generalization reflects the most realistic scenario especially when working with city center like environments.

In practice, connected infostations are grouped in clusters. Depending on the types of connections present among the infostations, we might end up with a various number of clusters in the network. Infostations that belong to the same cluster are willing to either

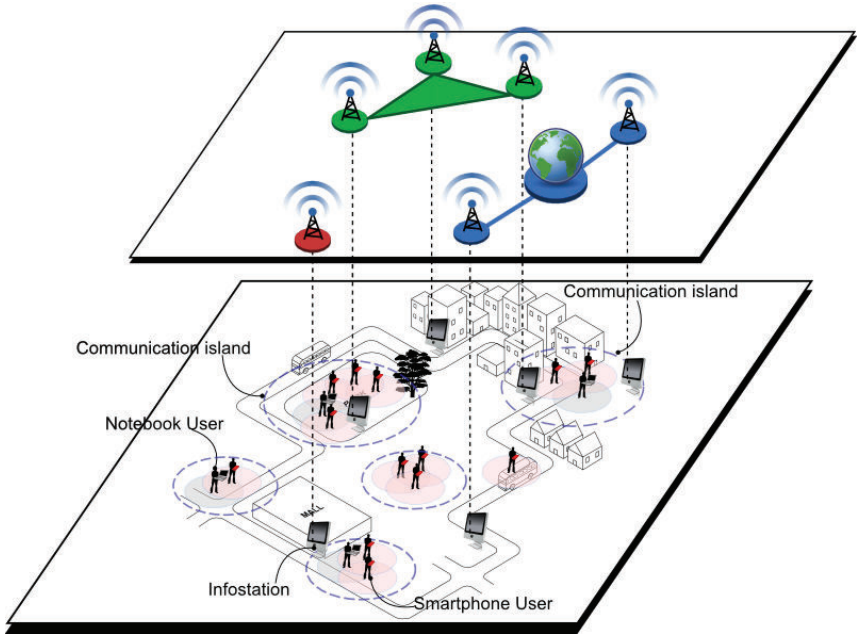


Figure 2.9: Example of ICHN with connected and isolated infostations

act as proxies to each other or route requests they receive to the desired infostation. Thus, a mobile client can send the service request to any infostation of a specific cluster, and not compulsory to the one providing the service. In other words, a client generates the service request and sends it in anycast mode. In contrast, if the infostation providing the service is isolated, the client should rely on unicast mode to send the service request.

2.2 Service-Oriented Computing

Service-Oriented Computing (SOC) is a computing paradigm that utilizes services as fundamental elements for developing applications. It is a programming approach that supports the dynamicity of the environment and the rapid and low-cost development of distributed applications in heterogeneous environments. The promise of Service-Oriented Computing is a world of cooperating services that are being loosely coupled to flexibly create dynamic processes and agile applications that may span computing platforms.

Service-Oriented Computing is a keystone of opportunistic computing. As defined by Conti, in OC each user can avail not only of the resources available on its own device, but can also opportunistically leverage on other resources of the environment, including those on other users' devices, in a trustable and secure way. Users can compose the functionality of the different resources available in the network, enjoying much richer functionality than that available on their own devices. Opportunistic computing thus generalizes the concept of opportunistic networking by considering the opportunistic use of any resource available in the network.

The following part is devoted to introduce the service-oriented approach and present its principles. We will also introduce the Service-Oriented Architecture (SOA) that is used to design an application as a set of services, showing the convenience of dealing with the

available resources (hardware or software) in the environment as services.

A **service** is a piece of software that provides a set of functions that can be requested by other devices for usage. It is a function provided by a supplier, or a service provider, to a client who is the consumer of this service. This function defines the interaction pattern that must exist between the client and the service provider. In order to represent the client and the provider of a service we will use, in the remainder of this thesis, the notation shown in Figure 2.10: examples of software services can be web services providing weather forecast, stock quotes or language translation. Services provided by hardware devices can be, for example, controlling a temperature in the room or printing a document.

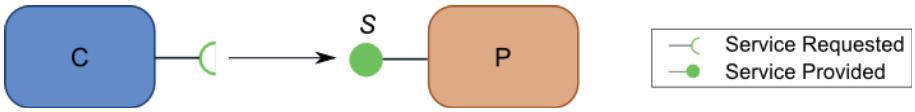


Figure 2.10: The client C of a service S is linked to provider P

The literature offers a wide range of definitions of software services. However, there is still no consensus on a single definition of a service.

"A service contractually defined behavior that can be produced and provided by any component to any other component, on the basis of the contract" [8].

"Service: The means by which the consumers' needs are related to the capacities of the supplier" [80].

We propose a more general definition of a service that takes into consideration the point of view of both the client and the service provider.

"A service is a unity of modularity and deployment. It is a specification of the features that can be supported by the service providers. The usage of a service by a client is defined by a contract that specifies both the functional and non-functional properties of a service".

In this definition we represent a service by its contract. The service contract includes both the functional (functions and data types) and non-functional (behavior characteristics) properties of the service. So as to interact, a client and a provider need to share a contract. The tasks performed by the service are the functional properties where the security or the quality of the service are non-functional properties. Non-functional properties allow a client to differentiate between two services similar from a functional point of view. The difference between the functional and non-functional properties of a service can be represented by taking the online printing of digital photographs as an example. Several providers can propose such a service. However, even if the functionalities are the same, each provider has his own way to perform such a process. These services will not be identical concerning several non-functional aspects: the price, the quality of the paper used, the time needed for processing and printing, the delivery tracking, etc. From that we can deduce the following equivalence:

"Two services provided by two service providers are considered equivalent if and only if their functional and non-functional properties are equivalent".

After presenting the definition of the service, we focus on some of the principles that define the service-oriented paradigm that were first introduced by Erl [28].

Establishing a *loosely-coupled* relationship between services is a main principle for the service-oriented paradigm. Loose coupling allows asynchronous evolution of clients and providers and important flexibility. Indeed, services run on different platforms are implemented independently and have different providers, thus they must not require knowledge or any internal structure of the client's side.

A service directory is introduced between the service provider and the client in order to ensure both the loosely-coupled and the late binding principles. Thus, to define the various parts that are represented in Figure 2.11:

- The *service provider*: The service provider is the part responsible of defining the service descriptions and publishing them to the service directory.
- The *service requester*: The service requester (also known as the client or consumer) wanting to use the services, accesses a service directory to find the service descriptions and their providers.
- The *service directory*: Sometimes called registry, or broker of services, plays the role of the central entity. Its intermediary role between the clients and the service providers which allows the decoupling and the late binding principles.

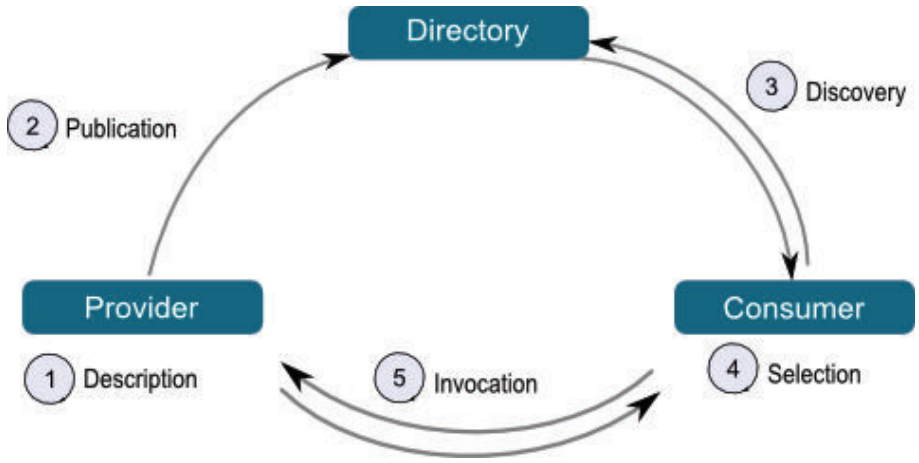


Figure 2.11: Interactions among the different entities in the network

Loose coupling also influences the *late binding* of clients to providers. Since a client is not bound to a specific provider, an invocation link is only created at execution time.

As described in Figure 2.11, the different activities that can be performed are: description, publication, discovery, selection and invocation. Before a service can be used, the service provider must be located by the directory. For that, service providers start with the description of the services i.e., specifying the contracts of the services they support. This step is followed by the publication of the service contracts to the directory that is in turn contacted by the clients to discover and select one or more services matching their specific interests. The client acquires the address of the provider of the desired service and only then the client can be linked to the service provider and use his service(s).

In order to provide a better selection mechanism of the services, it is important for services providers to include the non-functional properties in the contract beside the

functional properties of the service. Thus the client can, by relying on some filters (Figure 2.12), choose among several providers of the same service.

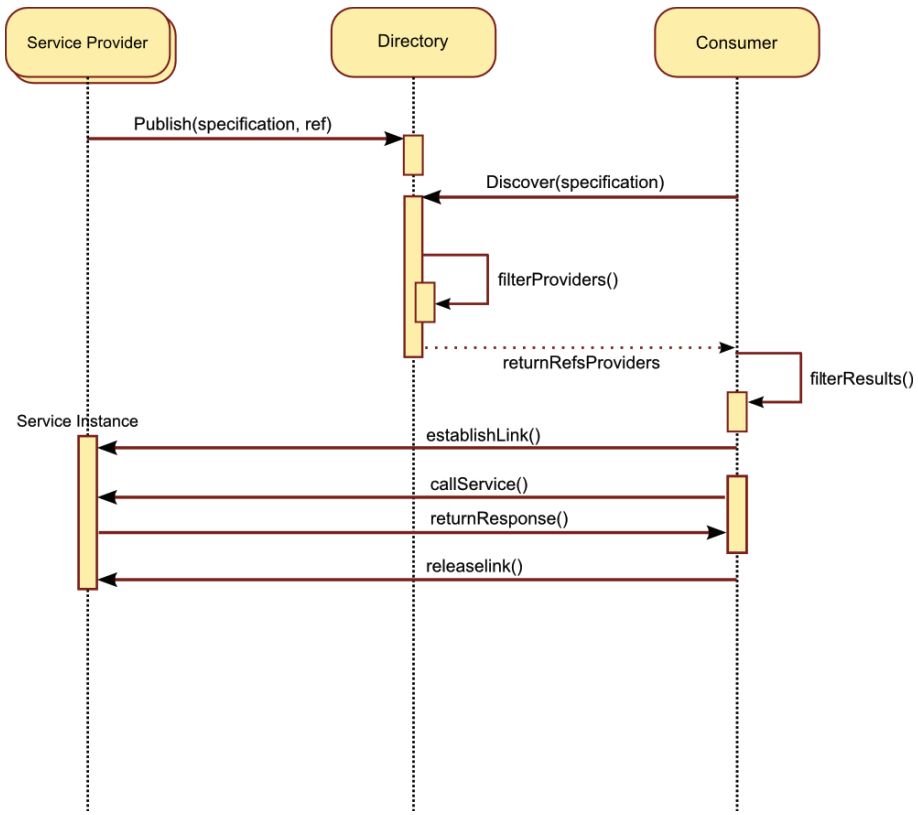


Figure 2.12: UML Sequence Diagram of the SOC

As for the directory, it can be unique and centralized, shared among several subdirectories distributed over the network, duplicated to overcome the faults, etc.

Service-oriented paradigm also encourages *reuse* in all services, regardless of whether immediate reuse opportunities exist, services must be designed to support potential reuse. Thus favoring the dynamicity that is needed for pervasive computing.

Services may *compose* other services. As depicted in Figure 2.13, this principle ensures that services are designed so that they can participate in numerous complex compositions of other services. Furthermore, the device of the client can also host a service provider. Indeed, by using one or more services, the client can create new functionalities and this propose new services. This is a form of service composition.

Service-oriented systems are characterized by high levels of scalability and flexibility due to the principles of the service-oriented design paradigm they are based on. In fact, building applications that take part in smart pervasive environments implies the integration of various heterogeneous communicating objects. Such kind of applications presents significant needs in terms of scalability and flexibility. Thus, this can be accomplished by accessing and controlling these objects using services.

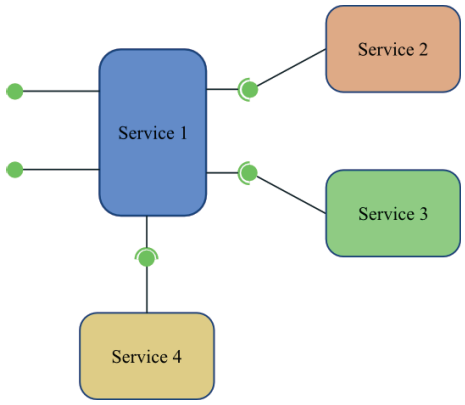


Figure 2.13: Composition of a new service out of several services

2.3 Service Provisioning Process

The basic service-oriented computing we have presented have been investigated in the area of ubiquitous and pervasive computing [98]. For example, in [106] Ravi et al. developed a mechanism for accessing pervasive services across the network of devices using cell phones. In this thesis, our main objective is tackling the service provisioning challenge in intermittently connected hybrid networks. In such networks, different nodes offering various services may enter and leave the network at anytime. Moreover, these devices can move at high or low speeds or even remain stationary. To benefit from the services provided by such devices, similar to the basic service-oriented architecture, a client must be able to both locate the services in the network and invoke them. Nevertheless, due to the constraints that control such networks, several limitations are applied over the basic service-oriented architecture leading to major modifications in the advertisement, discovery and invocations approaches that can be applied.

In the following section, we focus on the different activities that can be performed as a part of the service provisioning process and that are common with the actions performed among the entities of the basic service-oriented architecture and we address the various existing discovery and invocation approaches that play an important role in the service provisioning process in intermittently connected networks.

2.3.1 Service Discovery

Provision systems are primarily intended to support network-wide service discovery. Service discovery allows clients to automatically discover services with their attributes and to advertise the contract of the services they provide to the rest of the network. Performing such a process with high success rate, low discovery duration and limited network traffic represent a prerequisite for the good utilization of services in the network and for an efficient collaboration in ubiquitous computing environments. We introduce in the following section the two basic service discovery architectures already presented in the literature and the possible modes for service discovery.

2.3.1.1 Service Discovery Architectures

Depending on whether a directory exists or not, there are three basic architectures that a service discovery approach may adopt: *directory-less architectures*, *directory-based architectures*, and *hybrid architectures*.

Directory-based Architectures

In directory-based architectures, there are three possible roles for a mobile node. A node can be a service provider, a client or a service directory. Being the entity that stores information about the services available in the network, the directory can thus enable service discovery and invocation. In directory-based architectures, a directory can be implemented as *centralized* (hosted by a single node) or can be *distributed* among several nodes. Service providers advertise their services to the directory using a unicast message. Since clients are informed about the available services in the network only through directory nodes, a client interested in a service should first contact a directory to obtain the service description, which is then used to interact with the service provider.

Directory-less Architectures

Directory-less architectures differ from the previous in that there are no service directories to mediate communication between service providers and service requesters. Basically, service providers broadcast advertisements and service requesters broadcast service discovery requests. A service provider, upon the reception of a service request, would generate a response containing the desired service and send it to the client.

Two possible service discovery modes are available for clients in the network to acquire information about the services, which are *pull* and *push* based service discovery modes.

- Pull-based Service Discovery Mode: In this mode, service providers do not distribute any advertisement of the services they offer, but clients on the other hand issue "service-searches" of the services they desire.
- Push-based Service Discovery Mode: In this mode, clients do not generate any service-searches, but service providers advertise their services on discrete time intervals. These advertisements are subjected to several limitations, such as limiting their forwarding to bound ranges instead of flooding the whole network.

2.3.2 Service Selection

Following the service discovery process a selection phase may precede the invocation process. When the opportunity is given to a client to choose among several providers of the same service, the client can prefer one provider over the other. In this thesis, we consider that the client performs a manual selection of the desired services and we do not go into the details of this process.

2.3.3 Service Invocation Communication Models

The selection phase of the service provisioning process is followed by the service invocation phase. In the service invocation phase a client actually interacts with the provider of a previously discovered service. This interaction is performed by a destination flow of information between the client and the service provider.This communication model is known as the destination-based communication model.

After the discovery and selection phases, the discovery reply obtained by the client contains the address of the service provider(s). The invocation process is thus achieved using a unicast, anycast or multicast communication between the client and the service provider(s), depending if two or more service providers are connected together thus it is enough to target one of them by relying on anycast, or if the targeted service requires contacting more than one service provider thus ending up with a group-based communication model (i.e, relying on multicast communication). Consequently, the invocation mechanism has been formulated as a routing problem of messages between clients and service providers.

2.4 Conclusion

Performing a scalable and efficient service provisioning in an ICHN is a difficult problem. In order to achieve this objective, we should overcome the challenges imposed by both intermittently connected networks (ICNs) and by service provisioning. This requires tackling specific patterns of communication for performing both service discovery and service invocation. Furthermore, there is the need for minimizing the message overhead for both service discovery and invocation while maintaining limited delays and overcoming intermittent disconnections.

In general, for service discovery, the challenge is viewed as a broadcast problem in ICHNs. Indeed, the main objective is introducing a dissemination mechanism to inform all the mobile clients of the provided services in the network.

As for service invocation, the communication pattern is a request/response relation between a mobile client and one (or more) service providers. The main objective can be summarized by performing a successful exchange of service requests and responses between a mobile client and the providers of the desired services. Thus, the challenge can be viewed as a unicast problem in ICHNs. Depending on the structure of the fixed part of the network in the ICHN (connected with a backbone or more than one providers acting as proxies of the original provider), this challenge can be also viewed as an anycast problem in ICHNs.

The various principles supported by SOC makes it interesting to be combined with opportunistic communication. For instance, the late binding and the loosely couple principles supported by SOC makes the service invocation possible after the service discovery phase. As it is mandatory to support asynchronous communication in challenging networks like ICHNs. Furthermore, SOC, by its various principles, provides the dynamicity needed by a client for the exploitation of several service providers. By that, we can say that SOC is a good paradigm for ICHNs.

Finally, in this chapter, we presented the challenging points that must be addressed for the design of a service provision platform that remains viable in disconnected mobile environments. We traced the evolution of wireless networks and introduced the service-oriented computing paradigm that is considered the keystone of opportunistic computing. The following chapters will present the state of the art covering the various research works in the communication domain of ICNs and the various approaches of service provisioning mainly in MANETs and ICMANETs.

Part II

State of the Art

3

Communication in Intermittently Connected Networks

In this chapter, we introduce the different approaches that aim at providing efficient communication in intermittently connected networks (ICN). Starting from communication in Delay Tolerant Networks, we introduce the various proposed protocols for ICMANETs and classify them according to the heuristics they rely on to perform their decisions. Then we give an overview of the projects that have been deployed in several real case scenarios.

3.1 Delay/Disruption Tolerant Networking

Today, the Internet has become an important, cheap, and widely used means of communication. The Internet Protocol (IP) is the primary protocol that establishes the Internet and is responsible for delivering the packets from a source to a destination. One of the most basic requirements for 'traditional' (IP) networking is that, the end points must be fully connected for the duration of a communication session (case of a TCP). Nevertheless, there are a number of locations, scenarios and situations where connectivity is intermittent, but the communication between nodes is still required. Indeed, there are many communities in the world where network infrastructures are not deployed (for cost reasons), such as the Sami in the north of Sweden or the tribes in the villages in Africa.

The past decade has seen significant research in the field of communication in challenged networks [30]. Protocols try to enhance traditional routing for connected MANETs by tolerating disconnections. These enhancements consist of enabling some or all of the network nodes to temporary store messages, in order to resend them later on when conditions permit. This type of networking is called Delay Tolerant Networking (DTN), and it is federated by the DTNRG working group [1] of the Internet Research Task Force (also known as Disruption Tolerant Networking, due to the support of the US Defense Advanced Research Projects Agency (DARPA), which has funded many DTN projects). DTN is a term invented to describe and encompass all types of long-delay, disconnected, disrupted or intermittently connected networks, where mobility and outages or scheduled contacts may be experienced [29].

The DTN architecture consists of independent networks each characterized by Internet-like connectivity within, but having only occasional communication opportunities among them. Such communication opportunities can be either scheduled over time or com-

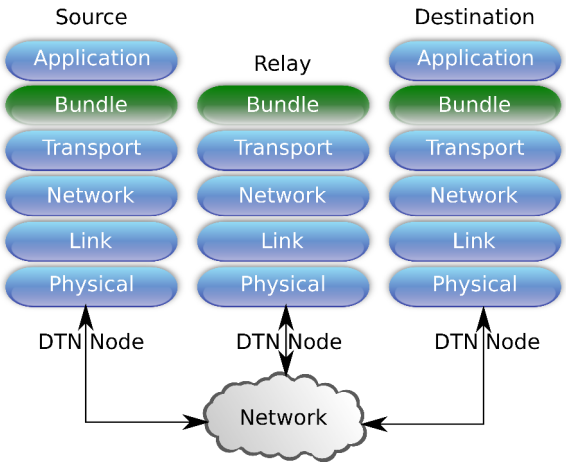


Figure 3.1: DTN Architecture and the Protocol Stack

pletely random. In general, the layered DTN architecture is based on the Internet TCP/IP five layered architecture and consists of an overlay, called the bundle layer, which operates above the transport layer, as depicted in Figure 3.1. The architecture specifies the format of variable length Application Data Units, called Bundles. The goal is to deliver Bundles from a sender to a receiver in the presence of intermittent and opportunistic connectivity, possibly over a wide range of different networks using different transport protocols. This is achieved by assuming that nodes store and forward bundles to cope with link disruptions. In the Internet architecture, TCP provides reliability to end-to-end communications. In the case of DTN, the bundle layer delegates on the lower layer protocols to ensure reliability in the communications. Thus, the bundle layer can be seen as a surrogate for the end-to-end path for intermediate nodes.

The Bundle Protocol [112] is the protocol that implements the bundle layer in the DTN architecture. After being designed solely for use in deep space for the "Interplanetary Internet", the bundle protocol has been proposed as the one unifying solution for disparate DTN networking scenarios including undersea networking, tactical military networks, ad-hoc networks, and other challenged networks. It is intended to consist of a group of well-defined protocols that, when combined, enable a well-understood method of performing store-and-forward communications.

Nevertheless, the Bundle Protocol alone does not solve the problems of networking in any of these environments. Rather, it is intended to provide a common format for store-and-forward networking messages and proposes that the availability of in-network storage in bundle agents (i.e., mobile devices in the network) will allow the challenges of these networks to be overcome. Many of the innovations that enable and support delay-tolerant network services should be understood as existing outside the basic Bundle Protocol itself, and as being largely independent of the Bundle Protocol. Indeed, the Bundle Protocol always sits upon a local transport 'convergence layer' whose design matches local network conditions. Bundle Protocol identifiers for routing, the Endpoint Identifiers (EIDs), are somehow mapped to local routing addresses in the local subnetwork via late binding. For that, the Bundle Protocol requires significant supporting infrastructure to function, in convergence layer adapters, and in enhancements or additions for specific implementations or deployments.

In reality, several deployments of the Bundle Protocol already exist. As said before, the DTN concepts came from Interplanetary communication. The NASA Scan program[1] and the European Space Agency (ESA) conducted a set of experiments to verify the feasibility of the Bundle Protocol among spacecraft and devices on Earth. The first feasibility experiment of DTN for space communication was presented in [133]. In 2008, Wood et al. used the Bundle Protocol in the Disaster Monitoring Constellation (DMC), a set of satellites that form a low-earth-orbit sensor network that captures images and sends the data payloads to a ground station. They employed the reliability of the Bundle Protocol to mask long delays and interruptions and highlighted the drawbacks of the DTN implementation. More recently, Jenkins et al. [50] described the implementation and tests with the Bundle Protocol in the International Space Station (ISS). They customized the publicly available ION implementation[2], which is a DTN implementation for embedded devices, in order to support long term delays and interruptions in remote commands among spacecrafts and the Earth. The authors concluded that the two main issues for the success of DTNs in space communication are selective acknowledgments and to improve the custody transfer algorithms. The Bytewalla project aims to provide Internet access in remote regions, such as villages with poor (or any) communications infrastructure in Africa [95]. The main idea is that travelers become "data mules". So, when users travel to a city with a better infrastructure, they carry the requests and after coming back to the village with the responses. To achieve this goal, they implemented a DTN2-based server, which follows the specifications of the Bundle Protocol, as well as a DTN-Email application for Android phones [96].

Where disconnections and delays in Interplanetary networks are due to the distance among source and destination, communication in opportunistic networks located in urban areas share similar disconnections and large delays. Indeed, opportunistic networks are networks where the low density and the sparse or irregular distribution of the nodes together with the harsh nature of the wireless links, may create long periods of node disconnections as well as partitions among sections of the network.

DTNs operate over the TCP/IP protocol stack, serving as a "gateway" for interconnecting Internets over delay and disruption-constrained links. An Opportunistic network is considered as a subset of DTN.

In fact, opportunistic networks focus on the the disconnection and interruption of communication among nodes within the same network. Moreover, they do not mandate the use of TCP/IP protocol stack, and are characterized by the use of the "store, carry and forward" paradigm. Thus, a device stores the messages in its local cache and waits for a contact opportunity with other devices to forward the messages. Upon contact, the device evaluates its neighbor as a good forwarding opportunity or not by relying on a set of information exchanged between the two devices. This evaluation is followed by conditional forwarding of the messages to the new neighbor. Finally, in opportunistic networks each single node acts a gateway. Opportunistic networks call for a radical revision of legacy routing approaches designed for the Internet or for well-connected MANETs that tend to perform poorly under long delays and frequent disconnections.

[1] http://www.nasa.gov/scan
[2] http://ion.ocp.ohiou.edu

3.2 Opportunistic Networking

Besides allowing nodes that are not connected at the same time to the same network to communicate with each other, opportunistic networks are also a possible way to improve the capacity of multi-hop ad hoc networks beyond the well-known theoretical limit found by Gupta and Kumar [38]. Actually, Grossglauser and Tse have shown that an opportunistic network in which nodes act as carriers can achieve constant capacity, irrespective of the number of nodes in the network [36].

Consequently, the opportunistic networking paradigm is one of the most innovative generalizations of the MANET paradigm. Indeed, while MANET represents an approach to mask the node mobility by constructing "stable" end-to-end paths as in the wired Internet, opportunistic networks do not consider the node mobility as a problem but as an opportunity to exploit. Thus, when a node does not have a good next hop to forward the data, it simply stores the data locally without discarding it, as would occur in a MANET. In addition, with the opportunistic paradigm, data can be delivered between a source and a destination even if an end-to-end path between the two nodes never exist by exploiting the sequence of connectivity graphs generated by nodes' movements. Such types of challenged networks can be used in many urban areas when it comes to providing people with information services without relying on complex and costly infrastructure. For example, providing nomadic people in a city center with some touristic information about the city, some news or traffic information or some advertisements about the various shops with the services they provide. Such information can be disseminated among the various devices carried by pedestrians by relying on the "store, carry and forward" principle, thus relying on the opportunistic contacts of these devices when they enter the transmission range of each other. Another example is deploying a simple communication system based on IEEE802.11g on airplanes (limited to 20 km communication link) in the aim of providing a cheaper alternative to other communication systems. These contact opportunities between airplanes are used to deliver on-flight generated data (either from passengers or company) to the ground or even backup data when no other communication systems are available [85, 86].

Several approaches have been adopted to achieve reliable communication in such networks. Indeed, routing techniques in ICNs typically aim to maximize the probability of message delivery. This probability is measured by the *delivery ratio* defined as the ratio of the total amount of data eventually arrived to destination to the total amount of data injected into the system. Another routing objective is minimizing the delay that each message experiences during delivery. It specifically consists of the time between the message injected by the source node and when its received by the destination node.

Flooding the message in the network seems to be the most trivial approach. Attempts to offset the limitations of flooding based approaches have introduced a host of other schemes. Some of them use knowledge of the history of contacts made by the nodes, to route messages. Some other algorithms forward messages to another node which has better probability to make the message move closer toward the destination. In fact, most of these approaches agree that the most successful routing approach in ICNs in the per-hop routing that strives to find a route on a hop-by-hop basis, i.e., by searching the most appropriate next-hop node only once having traversed the hop before. This approach has the advantage to utilize up-to-date information. In practice, routing performance

improves when more knowledge about the future topology of the network is available and exploited [48]. Unfortunately, this kind of knowledge is not easily available and a trade-off must be met between performance achievement and knowledge requirement.

In the following section, we discuss the evolution of the forwarding protocols that are specifically designed to function in ICMANETs. We classify these protocols based on the number of message copies generated and forwarded in the network and on the heuristics they rely on to choose the next carrier(s) to ensure successful message reception by the final destination. In fact, the decision of the best next carrier(s) and the number of replicas generated for each message is directly related to the amount of knowledge a node can collect about the current topology and state of the network. Such knowledge can be acquired by performing an excessive information exchange between the different nodes roaming the network, besides registering the various events a mobile node might encounter (e.g., history of contacts, visited places, inter-contact times, battery levels, etc.). For that, a compromise is usually applied between the amount of collected information and the blindness of the routing protocol when taking decisions concerning the number of replicas of each message and the best next carrier(s). For that, we classify the protocols into: delegation-based, content-based, replication-based and mobility-based routing protocols.

3.2.1 Delegation-Based Routing Protocols

Delegation-based routing protocols keep one copy of a message in the network and attempt to forward that copy toward the destination at each encounter. Delegation-based strategies for DTNs follow a different approach, they aim at offering a way to enhance the end-to-end reliability in opportunistic networks by moving the responsibility for reliable delivery of a message toward its final destination using the **custody transfer** [31]. Among possible contacts, a node delegates the message to the best possible carrier node to move the message closer to its final destination.

Spyropoulous et al. [117] proposed a set of single-copy based protocols for opportunistic networks. In the **Direct transmission**, a node forwards a message only if it contacts the destination. This approach has unbounded delay but each message is only transmitted once. In **Randomized Routing Algorithm**, the message is forwarded based on a probability value. This routing approach is only convenient for environments where node mobility is well known or highly predictable. Although, custody transfer has the advantage of producing very low network traffic, but if a delegated carrier node fails communication failures are inevitable.

In [83], Martí et al. propose **Time To Return (TTR)** routing protocol specifically designed for disaster scenarios. The protocol takes advantage from the fact that all medical personnel in an emergency scenario are coordinated by a leader that assigns actions and maximum time to return to the base for security reasons. Each node has its own time to return (TTR) value that is used for performing forwarding decisions. Upon contact with a node with a less TTR value, the node relays all its messages to this node. If messages are successfully relayed, the sender empties the buffer in order to keep only one copy throughout the network. In addition, a user can decrement the TTR value of a decision is made to return sooner than predicted to the coordination point.

3.2.2 Content-Based Routing Protocols

In recent years, publish/subscribe systems have been widely adopted in the context of wired networks and the Internet [99]. These systems generally include an infrastructure broker mechanism that is responsible for delivering relevant publications to interested subscribers. Such an approach is however not suitable for mobile wireless scenarios where fixed infrastructure cannot be assumed.

Generally, in content-based networking, information flows towards interested receivers rather than towards specific destinations. A few works have addressed publish/subscribe communications in the context of Delay-tolerant networking. In [24], Costa et al. present **SocialCast**; a content-based routing scheme for publish/subscribe communications in a DTN environment. In SocialCast, a publisher originally delivers a fixed number of copies of the message to carrier nodes. A message carrier will deliver a copy of the message to subscribers it meets or will delegate the message to another node that is selected as a more feasible carrier. The carrier selection is based on a comparison of utility values that reflect the probability of node to be co-located with another node that is interested in the message. In [135], mobile nodes run a community detection algorithm and in each community, the nodes with the highest closeness centrality (i.e., the shortest path to all other nodes in the community) act as message brokers. Nodes publish a message by delivering it to a broker in their community. Broker nodes in different communities exchange messages and if a broker receives a message that matches an interest of one of its community nodes, the message is flooded among the community. Another communication middleware system that supports content-based dissemination is **DoDWAN (Document Dissemination in Wireless Ad hoc Networks protocol)** [54]. This protocol relies on a different approach from those mentioned before. Instead of attempting to construct and maintain a routing structure, DoDWAN implements a selective version of the epidemic routing model proposed in [123]. It provides application services with a publish/subscribe API. When a message is published on a host, it is simply put in the local cache maintained on this host. Each host periodically informs its neighbors about the documents it is carrying and that match their interest profiles. Afterwards, each radio contact with another host is an opportunity for the DoDWAN system to transfer a copy of the message to that host whenever it is interested. A host can request the transmission of a document it is actually missing. Thus, no document is sent in the network unless it has been explicitly requested by a client host.

The scheme proposed by DoDWAN is more suitable to ICMANETs as it differs from the SocialCast routing scheme where the content can only be obtained from a limited set of message carriers, and from the solution proposed in [135], where the brokers push messages to the nodes in the network.

3.2.3 Replication-based Routing Protocols

Routing techniques based on data dissemination and replication perform delivery of a message to destination by simply diffusing it all over the network. The heuristic behind this replication policy is a compromise between the knowledge of possible paths toward the destination node of a message by the choice of an appropriate next-hop and the dissemination of the message everywhere. Based on the level of knowledge collected,

replication-based protocols can be divided into the different subcategories presented in this paragraph.

3.2.3.1 Flooding-Based Routing Protocols

Replication-based protocols insert multiple-copies, or replicas, of a message into the network to increase the probability of message delivery. Essentially, these protocols leverage a trade-off between resource usage (e.g., node memory and bandwidth) and probability of message delivery, where the number of replicas in the system directly dependent on the number of nodes in the system. One of the first routing protocols in this domain is the **Epidemic Routing** protocol [123], which can, in a way, be assimilated to a simple flooding. According to the Epidemic Routing protocol messages diffuse in the network similarly to diseases or viruses, i.e., by means of pair-wise contacts between individuals/nodes. A node is *infected* by a message when it either generates that message or alternatively receives it from another node for forwarding. The infected node stores the message in a local buffer. This approach is utilized when the node has absolutely no knowledge about the network, where this knowledge helps in deciding the best next hop. As a result, when a message arrives at an intermediate node, the node floods the message to all its neighbors (except the one who sends it). To identify if the neighbor has already seen the message, each node maintains a summary vector. This is an index of the messages that its has already seen. When two nodes meet, this summary vector is exchanged. This enables the nodes to identify the new messages and request them. Undoubtedly, flooding the network with messages will guarantee the shortest possible delay in distributing them through the connected portions of the network, but since no precautions are taken to limit the number of messages exchanged and forwarded, this approach is very resource hungry both in terms of memory occupancy and bandwidth usage and, as a result, lead to high energy consumption and poor scalability. As demonstrated by Tseng et al. [93], this can seriously degrade the performance, especially in high-density regions or if the resources are scarce.

3.2.3.2 Random-Based Decision Protocols

In the aim of limiting the number of messages generated by the Epidemic routing protocol, some approaches perform a random dissemination of limited number of messages instead of flooding the network. For example, the **Spray and Wait protocol** [118] is inspired from flooding-based delivery schemes, in that, it makes no use of information on network topology or knowledge of the past encounters of nodes, however, it significantly reduces the transmission overhead by limiting the total number of copies that can be transmitted per single message. Spray and Wait breaks routing into two phases: a *Spray* phase and a *Wait* phase. In the *Spray* phase, L copies of a message are spread in the network from a source to L distinct nodes. This is followed by the *Wait* phase where the L nodes carry the message copy and use *direct transmission* to forward it only to its destination. Despite that, this approach succeeds in limiting the overhead of the flooding-based protocols since the number of messages forwarded on the network is limited to L, but it suffers from sever delivery ratios coupled with large latencies.

3.2.3.3 Partial Context-Based Decision Protocols

Partial context-based protocols act as a compromise between Flooding-based and Random-based decision protocols. They are called partial context-based since they rely only on the information gathered from the contacts between the nodes to perform the decision of the next carriers. Extra information might be also inferred from the mobility of the nodes. However, these protocols discard any information related to the profiles of the people carrying the devices.

Among partial context-based approaches are the protocols that rely on history of encounters between nodes or even history of visits to locations to perform their selection of next carriers. For example, in **Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET)** [77], the routing decisions are taken according to delivery probabilities that are estimated from the frequency of encounters. Thus, it attempts to reduce the overhead of the Epidemic protocol by identifying the mobility or the contact patterns. In fact, PRoPHET assumes that node movement is not random and that it is possible to identify mobility patterns. If a node has visited frequently a part of the network, there is a high probability that this same place will be visited in the future. For that, each node holds a delivery predictability table with the probabilities of successful delivery towards each known node in the network. Thus, the probability to deliver a message to a certain destination node increases whenever it comes within sight, and decreases overtime in case no meeting occurs.

A similar protocol that controls flooding by predicting nodes movements based on history information is the **MaxProp** protocol [11] designed for vehicle-based DTNs. Such a DTN is characterized by large storage capacity and energy source, but short contact duration. Hence, MaxProp discusses the prioritization of packets to be forwarded and dropped. These priorities are decided according to two strategies depending on the hop counts values of messages. If the hop counts of a set of messages is less than a predefined threshold, then messages with lower hop counts are given a higher rank. On the other hand, if the value of the hop counts is higher than the threshold, the ranking of the messages is based on the delivery likelihood. Where the deliver or path likelihood is the cost based on an estimation of the route failure likelihood calculated from the history of encounters between nodes. Packets that are ranked with highest priority are the first to be transmitted during a transfer opportunity. Packets ranked with lowest priority are the first to be added to make room for an incoming packet.

PropTTR and PropNTTR [84] are a follow-up of the TTR protocol and likewise designed for emergency scenarios. To overcome the moderate message delivery ratio of TTR resulting from the one copy approach that produces a lot of lost opportunities, PropTTR is based on both TTR and MaxProp. It uses MaxProp protocol for the first hop of the message (i.e., forwarding messages with a hop count of 0 based on the delivery likelihood approach of MaxProp) and TTR for the messages of a hop count > 0. By that, if a node meets c one-hop neighbors a maximum of c + 1 copies of a message will be forwarded. PropNTTR follows the same rules as PropTTR but instead of changing the forwarding decision algorithm when the message hop count = 1, the change is performed when the hop count of the message is equal to N. PropNTTR is mainly used in scenarios with low density of nodes.

Another protocol that benefits from context information is the **Context-Aware Rout-**

ing protocol (CAR) [89]. CAR attempts to predict if a destination belongs to the same connected part of the network of the sender. It uses the classical routing protocol DSDV (ad hoc routing) to provide message forwarding inside the connected clouds of a partitioned ad hoc network. Otherwise, it employs a unity function based on the change rate of the connectivity and the probability of the destination to be in the same cluster of the relay node. In other words, CAR focuses on expanding the classical routing table of nodes to support forwarding across disconnected ad hoc clouds by exploiting some contextual properties using utility functions and Kalman filters for computing the message delivery estimation based on the number of neighbors, the energy levels and the degree of mobility.

The authors of Spray and Wait introduced a follow-up of the protocol called **Spray and Focus** [119]. Spray and Focus uses a Spray phase similar to that introduced in Spray and Wait, followed by a focus phases. Unlike Spray and Wait, where in the Wait phase messages are routed using direct transmission (i.e., forwarded only to their destination), in the Focus phase a message can be forwarded to a different relay according to a given forwarding criterion. Specifically, a node forwards its messages to a better carrier selected according to a set of timers that record the time elapsed since two nodes last encountered each other. Although this kind of approach efficiently reduces the network load, the custody transfer implemented in the second phase tends to excessively delay the delivery of messages in some scarcely populated networks.

In **MobySpace Routing** [69], the nodes' mobility pattern is the context information used for routing. The protocol builds up a high dimensional Euclidean space, named MobySpace, where each axis represents a possible contact between a couple of nodes, and the distance along an axis measures the probability of that contact to occur. Two nodes that have similar set of contacts, and that experience those contacts with similar frequencies, are close in the MobySpace. The best forwarding node for a message is the node that is as close as possible to the destination node in this space. Obviously, in the virtual contact space just described, the knowledge of all the axes of the space also requires the knowledge of all the nodes that are circulating in the space.

3.2.3.4 Full Context-Based Decision Protocols

Another set of protocols go beyond context information that can be exported from the frequency of meetings and history of contacts, to analyzing information based on the habits of the users and their relationships among each other. Thus, they are called full context-based since they combine both the contacts between nodes and the profiles of the users carrying the devices.

A **history based forwarding protocol for opportunistic networks (HiBOp)** was proposed in [9, 10]. HiBOp utilizes the network topology, the history of contacts and user context to allow a node to learn its similarities with other nodes, so a message is spread among similar users. HiBOp requires users to provide information about themselves, like home and work address, hobbies and fun, etc. The basic idea is that a node with similarity information to that of the destination has more chances to meet the destination. As a consequence, HiBOp stores a contact history with the place that nodes last met each other to aid in forwarding.

In [92], Nguyen et al. propose a context-based forwarding protocol called **Propicman**. In Propicman, the context information of each node is represented by means of a node profile with evidence/value pairs. To each evidence is associated a weight that represents the importance of this evidence in the network. Thus, when a node wants to send a message to a specific destination, it sends to its two-hop neighbors the information it knows about this destination. Based on this knowledge, the neighbor nodes compute their delivery probability and return it. The source node then forwards the message to the neighbor node with the highest delivery probability. Moreover, the authors propose a security solution based on public hash functions to enable partial matches at intermediate nodes while allegedly protecting privacy of the destination.

In [45], Hui et al. show that it is possible to detect characteristic properties of social grouping in a decentralized fashion from a diverse set of real world traces, and demonstrate that such characteristics can be effectively applied in message forwarding decisions in DTNs. The authors propose **BubbleRap**, a forwarding protocol which uses social network metrics to choose which nodes should relay a message. Concretely, based on observations that human interaction is heterogeneous both in terms of popular individuals and groups pr communities, Hui et al. propose a social based forwarding algorithm. First, they propose a distributed method to detect communities. Thereby, each node knows its community. Then, they use the *betweenness centrality* metric as a rank of each node. Thus, when a node has a message, it *bubbles* the message to a node with a higher rank.

SimBet [25] is a forwarding protocol based on social interactions that uses centrality and social similarity metrics to define the probability of the node contacting the destination. In this context, nodes with high centrality values will be bridges between different communities in the network. Further, nodes with higher similarity indexes have the highest probabilities to find a common neighbor with the destination. Thus, first the authors use the centrality metric to exchange the message among the communities and net the similarity metric is used within the community to deliver the message to the destination.

Li et al. [72] proposed the **Social Selfishness Aware Routing (SSAR)**, a forwarding protocol that employs social characteristics to define if the node should forward a message. The authors define social selfishness as the willingness of a node to forward messages received from nodes with which it has no social ties. This contrasts with most protocols in the literature, which assume that the nodes will always forward the messages. Nevertheless, the protocol assumes that every node knows its social relationship with other nodes and allocates resources based on this knowledge.

Vu et al. proposed a forwarding protocol based on the observations of a real trace [127]. The authors collected Wi-Fi/Bluetooth traces of 123 phones during six month in 2010. They showed that the pattern of encounters between phones is regular and predictable. As a consequence, they proposed **3R**, which learns the patterns of encounters, creating a table with per-node encounter probability for each hour of each day (weekday and weekend). The main drawback of 3R s that it requires a long bootstrapping period.

3.2.4 Mobility-Based Routing Protocols

In mobility-based routing protocols, one or more stationary (or mobile) nodes are used to facilitate data transfer. Frenkiel et al. [32] was the first to propose the concept of info-

station, a low cost stationary node, to aid in message transfer in disconnected networks. Where, in general, infostations operate as mediators between the user mobile devices and an infostation center, on which a variety of services are deployed and executed. The infostation concept was further explored by Goodman et al. [35]. In this model users can connect to the network in the vicinity of infostations, which are geographically distributed through the area of network coverage. Infostations provide strong radio signal quality to small disjoint geographic areas and, as a result offer very high rates to users in these areas. However, this model only supports one-hop communication where a node must be in the infostations' coverage areas to transmit data, thus this lead to large delays.

Similar to the infostations, some devices play the role of mobile data collectors. They move around in the network, following either predetermined or arbitrary routes, and gather messages from the nodes they pass by. These special nodes are referred to as carrier, supports, forwarders, MULEs, or even ferries. They can be the only entities responsible for message delivery, when only node-to-carrier communications are allowed, or they can simply help increase connectivity in sparse networks and guaranteeing that also isolated nodes can be reached. In the latter case, delivery of messages is accomplished both by carriers and ordinary nodes, and both node-to-node and node-to-carrier communication types are allowed.

The **data-MULE system** [49] focuses on data retrieval from sparse wireless sensor networks and consists of a three-tier architecture. The lower level is occupied by the sensor nodes that periodically perform data sampling from the surrounding environment. The middle level consists of mobile nodes, called MULES, that circulate the area covered by sensors to gather their data. The upper level consists of a set of wired APs and data repositories which receive information from the MULES. They are connected to a central data warehouse where the data received is stored and processed.

In the **message-ferrying approach** [138], the non-randomness mobility behavior of specific nodes (the message ferries) is used to carry messages between the different partitions of the network with the aim of increasing the delivery rate and of reducing the message propagation delays. Two different instances of the Message Ferrying approach have been considered, both with movement constraints. The first approach is a node-initiated message ferrying where the ferry node moves around following a predefined and known path. Each node in the network has knowledge of the paths followed by active ferries, and is forced to move towards the ferry in order to send and receive messages. The second approach is a ferry-initiated message ferrying. Here, the ferry traverses predefined paths. Any source node wishing to deliver messages sends a service request to the ferry (via a longer-range radio signal), which also includes its current position. After having received the request from the source node, the ferry changes its trajectory to meet up with the source node.

Even if it is reliable, such an approach is not appropriate because it imposes too many constraints on mobile clients that should be able to move freely.

3.2.5 Communication in Presence of Infrastructures

The cooperation between wireless infrastructures and opportunistic networks has been investigated recently in order to enhance the content delivery to mobile clients and to

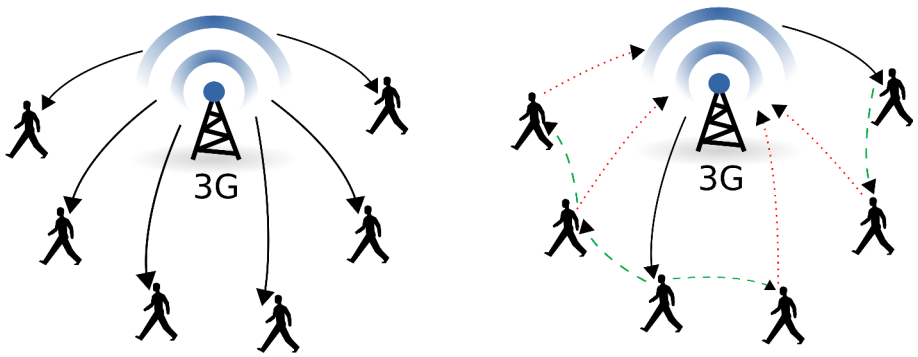


Figure 3.2: Push-and-Track Framework.

relieve the infrastructure.

For instance, some research works have focused on extending the coverage range of access points through opportunistic routing, such as the work proposed in [71]. The authors aim to extend the coverage of the existing Wi-Fi infrastructure through vehicle to vehicle (V2V) communications. They focus on an interaction between drivers that request some information from the access points (e.g., the status of the congestion on the highway) and asynchronously wait for the data. Thus, the response is routed opportunistically from the infostation toward the driver.

Other works focus on providing opportunistic communications among mobile users, such as the work presented in [121]. Trifunovic et al. propose an opportunistic network system for Android smartphones called **WiFi-Opp**. WiFi-Opp is an alternative way of enabling opportunistic networking based on current smartphones' communication features and APIs. WiFi-Opp leverages the mobile Wi-Fi AP feature (also known as tethering) as well as stationary open APs to support opportunistic communications between classical Wi-Fi clients. Thus, the authors enable opportunistic communication between the mobile nodes in the network by suggesting several modes of operation for the smartphones. For that, smartphones can scan the environment for usable access points and associate with them and optionally switching between them. If no access point can be found, a smartphone becomes an access point itself for a limited amount of time to facilitate the communication of other nodes.

In [51], Whitebeck et al. tackle the problem from another point of view. With their solution **Push-and-Track**, the authors aim at disseminating content to mobile nodes while meeting guaranteed delays and minimizing the load on the wireless/3G infrastructure. This framework consists of a control system which pushes content to mobile nodes and keeps track of its opportunistic dissemination. The framework relies on a close-loop controller to decide when to push new copies of the content through the infostations and to which set of nodes these messages should be forwarded to ensure a smooth and effective dissemination using epidemic routing. Thus, Push-and-Track favors the opportunistic ad hoc communication over the direct 3G connection between the smartphones and the 3G infrastructure whenever possible (Figure 3.2). Although the global infrastructure load is reduced by relying on this framework, relying on the epidemic routing protocol and requiring acknowledgments to keep the loop closed may significantly increase the network overhead.

In [97], Ott et al. present protocol mechanisms for running HTTP-over-DTN. They aim to overcome the unpredictable loss of network access for mobile users traveling by car, bus, or trains. The authors state that, such users usually experience varying connectivity characteristics while they are on their way, thus they operate in a challenged networking environment that is not suited for many Internet applications. The authors present a protocol design and system architecture for delay-tolerant access to web pages by running a slightly enhanced variant of HTTP on top of the DTNRG bundle protocol which enables web access for intermittently connected users.

3.3 Realistic Case Studies

Research on opportunistic networking is devoting particular attention to realistic case studies. In fact, the implementation of real-application scenarios in opportunistic is increasing. For example, wildlife tracking applications aimed at monitoring wild species in unmanned scenarios. In such a scenario it is important to limit human intervention in order to respect natural ecosystem, and thus it is necessary to utilize light networking. Another example of opportunistic application consists in providing Internet connectivity to rural and developing areas where conventional networks do not exist. Deploying traditional (wired or wireless) networks to cover these areas is not cost-effective, whereas opportunistic networks are an affordable solution.

3.3.1 Wildlife Monitoring

Wildlife monitoring is an interesting application field for opportunistic networks. It focuses on tracking wild species to deeply investigate their behavior and understand the interactions and influences on each other, as well as their reaction to the ecosystem changes caused by human activities. Researchers use opportunistic networks as reliable, cost-effective, and not intrusive means to monitor large populations roaming in vast areas.

In **ZebraNet** [53, 78], wireless sensor nodes, namely collars (attached to Zebras) equipped with GPS, a solar battery, memory, a small CPU and wireless transceivers, collect location data and opportunistically report their histories when they come in the radio range of base stations. The base station collecting the data consists of a mobile vehicle for the researchers, which periodically moves around in the savanna and collects data from the zebras encountered. Two alternative protocols have been considered for data collection in ZebraNet. The first one is simple flooding, where each collar sends all its data to each encountered neighbor until the data eventually reach the base station. The second one, named history-based protocol, proposes that each node selects only one of its neighbors as relay for its data. The selected node is the one with the highest probability to eventually encounter the base station. Each node is thus assigned a hierarchy level (initially zero) that increases each time it encounters the base station, and conversely decreases after not having seen the base station for a certain amount of time. When sending data to a relay node, the neighbor to be selected is the one with the highest hierarchy level. Experimental results indicate that the flood-based protocol yields higher system throughput if the buffer capacity at each node is large enough, but energy consumed by the flood-based protocol is very high compared to that by history-based protocol. There is a trade

off between throughput and energy consumption. Their conclusion is that while flooding makes sense at low-radio-range and low-connectivity points in the design space, it is not a good choice in a high-connectivity regime.

In the **Shared Wireless Infostation Model (SWIM)**, whales are the wild species to be monitored [116]. It is also a merging of the infrastructure-based routing and the epidemic routing protocol. Special tags applied to the whales perform periodic data monitoring. Data is replicated and diffused at each pair-wise contact between whales and finally arrives to special SWIM stations that can be fixed (on buoys) or mobile (on seabirds). Hence, both whale-to-whale and whale-to-base-station communications are allowed. From the SWIM stations, data are eventually forwarded onshore for final processing and utilization. One of the benefits of SWIM, by allowing the packet to spread throughout the mobile nodes, is that the delay for the replicas to reach an infostation can be significantly reduced. However, this comes at the price of consuming the network capacity.

3.3.2 Opportunistic Networks for Developing Areas

Opportunistic networks can provide intermittent Internet connectivity to rural and developing areas when they typically represent the only affordable way to bring the digital divide. One such example is the **DakNet Project** [100], which is aimed at realizing a very low-cost asynchronous network infrastructure so as to provide connectivity to rural villages in India, where it is not cost effective to deploy standard Internet access. According to DakNet project, kiosks are built up in villages and equipped with digital storage and short-range wireless communications. Periodically, mobile access points (MAPs) mounted on buses, motorcycles, or even bicycles pass by the village kiosks and exchange data with them in a wireless manner.

Another interesting opportunistic application scenario has been investigated in the framework of the **Saami Network Connectivity (SNC)** project [26], which aims to provide network connectivity to the nomadic Saami population of the reindeer herds. Saami herders live across the north part of Sweden, Norway, and Finland and move from their villages through the year following the migration of reindeers. Providing network connectivity to the Saami population is a means to protect and defend their habits, culture, and traditions while also supporting their integration into the modern society of their countries. With network connectivity Saami are allowed to continue to live according to their traditions and, at the same time, have much economic sustain through distance work and net-based business. Network-based services can also allow Saami children to receive their education without the need to leave their parents to attend boarding schools. In the initial state, the SNC project has only focused on providing email, file transfer, and cached web services to the Saami people. It should finally be noted that the SNC project focuses on a pure DTN architecture.

The **Networking for Communications Challenged Communities (N4C)** project shares the same objectives as the SNC project in extending Internet access to people, businesses and authorities operating in communications challenged communities. In cooperation with users in Swedish Lapland and the Kočevje region in the Slovenian mountains, the project conducted field trials of architecture, design, infrastructure and applications. The opportunistic networking architecture envisaged for N4C builds on the SNC architecture. The N4C project deployed real DTN systems in remote areas of the Swedish mountains

and as a result, it has allowed the Saami population to send email and access static web content over DTNs. Moreover, one component of the N4C system tested during annual summer deployments, was an implementation of PRoPHET.

3.3.3 Social Applications

MobileMAN and **Haggle** projects address the communication in presence of intermittent network connectivity using opportunistic networking techniques. MobileMAN aims at defining and developing a metropolitan area, organizing and totally wireless network, where the users' devices form the network, and no infrastructure is required. This project thus fostered a kind of "citizens network" by which people could avoid the operators infrastructure (i.e., communication costs). Haggle[3], which can be perceived as a successor of the MobileMAN project, aims at designing a new autonomic networking architecture to enable communication in presence of intermittent network connectivity, which exploits autonomic opportunistic communications (i.e., in the absence of end-to-end communication infrastructures). In this framework, researchers study the properties of Pocket Switched Networks (PSNs), that is, opportunistic networks that can exploit any possible encountered device (e.g., cell phones and PDAs that users carry in their pockets) to forward messages. The project proposes to completely eliminate layering above the data-link, and to exploit an application-driven message forwarding, instead of delegating this responsibility to the network layer. From this point of view, Haggle intends to define a communication layer implementing a best-effort message forwarding between mobile devices, in order to ensure content delivery when connectivity is local and intermittent.

Transhumance project aims at designing and at implementing a middleware platform for supporting the execution of collaborative data sharing applications in mobile ad hoc networks. Transhumance considers typically groups of less than 100 people that share multimedia documents during a game or a spontaneous meeting. The middleware platform is designed in order to support the mobility of users which are expected to move in a limited geographic area. Group-oriented services are distributed on the mobile devices. During field tests, Transhumance has shown that several mobile devices can communicate in ad hoc mode spontaneously, and can forward and share documents in a collaborative way.

Project **PodNet** [70] has objectives quite similar to those of Transhumance. PodNet aims at developing the system software and protocols to enable efficient wireless ad hoc content distribution on mobile devices. Moreover, it aims at understanding the social behavior of users and their mobility patterns while using such a system. PodNet is data-centric with content-based addressing and provides applications with group-based communication through publish/subscribe interface. Podnet relies on the opportunistic contacts between mobile devices to perform a peer-to-peer content distribution. It relies on a socially interacting and cooperating group of users. In general, transfer opportunities arise when people with matching interests meet such as in public transportations, at parties or urban areas. Thus upon pair-wise contacts, users exchange their interests and select which data objects to exchange.

[3] http://www.haggleproject.org

Another project that focuses on content-based dissemination in delay and disruption tolerant networks is the **SARAH** project. This project addresses the issues related to the design and utilization of distributed services on mobile devices. SARAH implements its own model for delay-tolerant ad hoc networking [54, 41]. It also investigates the possibility of discovering and invoking services using delay-tolerant communications by implementing a middleware platform that serves as a platform for this purpose [62, 81]. In contrast with most of the current works focusing on delay-tolerant networking, the SARAH project devises models based on content-based routing [42, 13, 14] and gossiping techniques in order to support message differentiation, and thus to help each device decide what messages it should accept to store, for how long, etc.

3.4 Discussion and Conclusion

In this chapter, we introduced the different categories of the routing approaches that can be deployed to function in DTNs and ICNs while taking into consideration the frequent disconnections in the communications in these networks. The classification of the protocols into various categories is based on the amount of knowledge needed by each protocol in order to perform routing operations. Such operations might vary from the choice of the next carrier(s) to the number of duplicates that should be generated.

In this classification process, we first introduced the delegation based routing protocols giving various examples. Despite that such an approach succeeds in reducing the network load, it fails in obtaining high message delivery ratios in challenged networks similar to those we deal with in our work. Another category is the replication-based routing protocols. This set of protocols deal better with disconnections due to the presence of multiple copies of the messages in the network. Among the well known approaches in this category are flooding-based routing protocols. Despite that these protocols outperformed the delegation based protocols in both delivery ratios and delay, they suffer from congestion due to the high number of messages generated. In order to avoid congestion, limiting the number of generated messages is inevitable. Protocols that rely on partial or full context information to perform this limitation were introduced. In these protocols, decisions concerning the number of messages to be generated and the choice of next forwarders are taken according to information collected from the network. The amount of collected information needed to perform successful routing of messages in the network critical. Indeed, in our work, the resources and the computational capabilities of the devices forming the network are scarce and limited. For that, a large number of the already proposed protocols choose to discard this fact. In the solutions we propose, we tend to minimize the amount of information and knowledge needed to perform successful routing decisions.

On the other hand, the majority of these solutions are not dedicated for service delivery. They focus on forwarding messages from one node to another without exploiting the client/server (request/response) paradigm. In our work, we tend to show the importance of designing a protocol that takes this communication pattern into consideration by comparing our solution to other general purpose protocols. Moreover, these solutions target networks either solely composed of mobile devices or involving both standalone fixed infostations and mobile devices, and thus do not benefit from the presence of fixed

and inter-connected infostations in the network. By taking this into consideration, several improvements over service delivery in ICHNs should be introduced. For example, providing better access continuity for nomadic people roaming the network can be accomplished by relying on a handover mechanism between the infostations that takes into consideration the opportunistic behavior of the people in the network.

4
Opportunistic Computing

In this chapter, we cover the various service provisioning approaches found in the literature. In the first section, we present the different approaches of service discovery and invocation in MANETs. In the following section, we follow the evolution of these approaches to meat the requirements of ICMANETs. In the final section, we give an overview of the handover mechanisms for service provisioning in both MANETs and ICMANETs.

4.1 Service Provisioning Systems in MANETs

MANETs actually cover a wide variety of situations depending on various parameters, such as the density, mobility, volatility of the nodes forming the network. For that, taking into account the characteristics of the network environments, service provisioning systems can be divided into two types: those designed for *stable MANETs*, and others designed for *dynamic MANETs*. In stable MANETs, the mobility of the devices is negligible or limited where such devices do not leave and join the network frequently. As for dynamic MANETs, the devices forming such networks are characterized as highly mobile and more volatile thus resulting a frequently changing network topology. This, in turn, has direct effect on the efficiency and performance of the various service discovery and invocation approaches. In this context, different approaches have been specifically designed for each type of network targeting the various challenges characterizing it.

4.1.1 Service Discovery in MANETs

4.1.1.1 Service Discovery Architectures

Before discussing the various approaches introduced for service discovery in both stable and dynamic MANETs, we introduce the service discovery architectures in such networks. As presented in Chapter 2, depending on the existence or absence of a directory in the network, three basic architectures of service discovery may exist: *directory-less architectures*, *directory-based architectures* , and *hybrid architectures*. Each of these categories can be divided into two or more subcategories as shown in Figure 4.1.

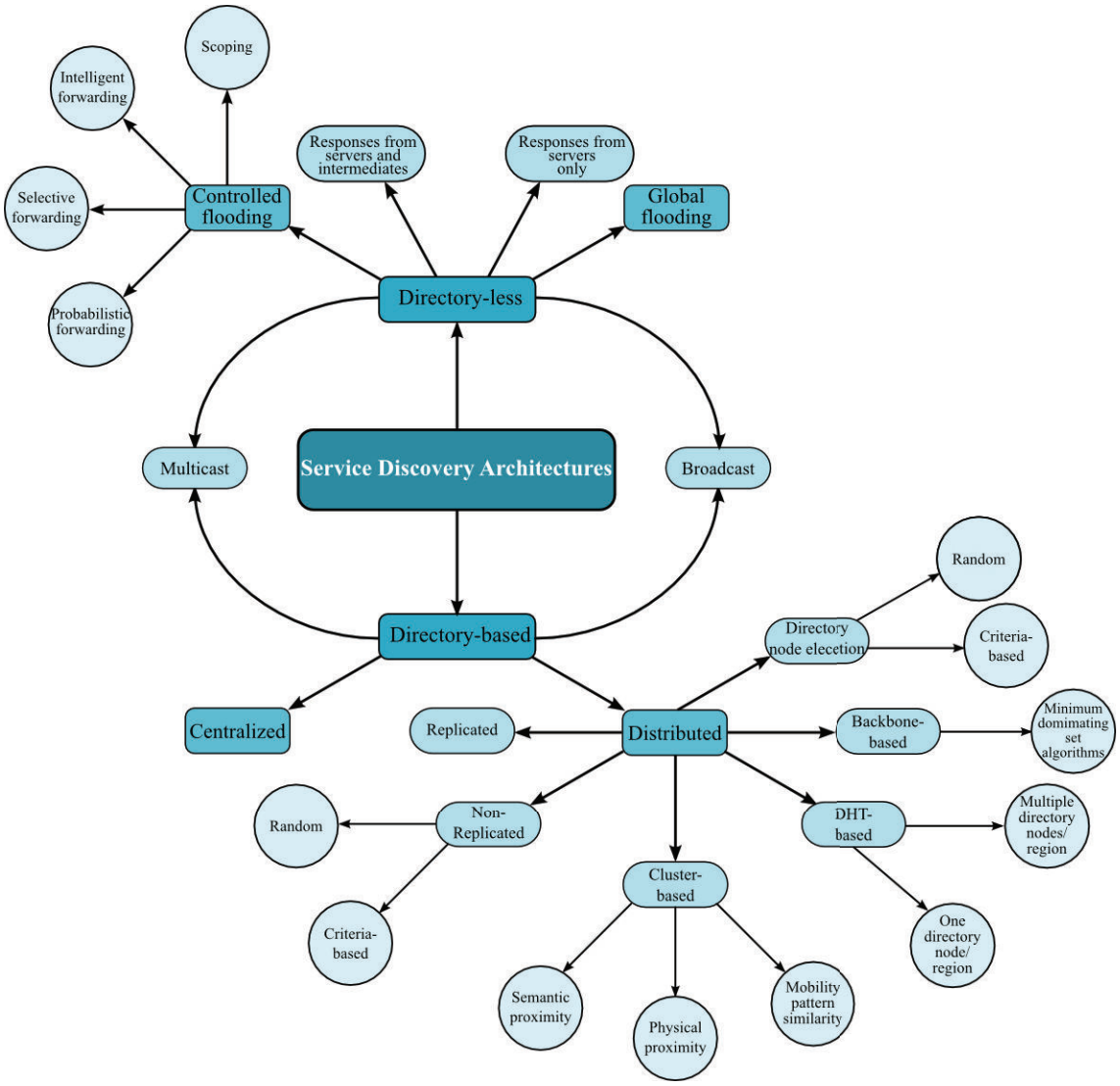


Figure 4.1: Service Discovery Architectures

Directory-based Architectures

A directory present in a directory-based architecture can be implemented as centralized (hosted by a single node) or can be distributed among several nodes.

Considering the centralized approach, the discovery of the directory by clients and service providers is in general based on multicasting. Service providers, upon discovering the central directory, unicast their services to the discovered directory. Clients, in turn, can obtain the service description, which is then used to interact with the service provider. Such approach was primarily adopted in wired networks and in wireless local area networks where one or more fixed hosts take up the role of a directory (e.g., UDDI [2]).

A basic question is whether global service discovery is to be provided (i.e., to be possible for every node to learn and invoke any service provided in the network). One ap-

proach is to use full replication of directories so that each directory stores all the available services in the network. A classical distributed directory approach is Jini [87]. In Jini, few nodes, named Lookup servers, act as directories. However, there is no communication among the Lookup servers. Thus, it is the responsibility of service providers to publish their services to more than one directory and keep them up-to-date. In this case, the automatic replication is not provided, global discovery is hence not supported, since services are advertised only in the area where the Lookup servers reside.

Other distributed directory approaches are based on protocols that create and maintain a backbone of directories. These directories are in constant communication with each other and are able to replicate and disseminate service information among each other. For example in [60], a backbone of directories is formed using a Minimum Dominating Set algorithm. Where in [109], a non random forwarding of requests among the different members of a backbone was proposed. It is sufficient for service providers to advertise their services to one directory of the backbone. So, a client and a service provider present in distant parts of the network can still discover each other and communicate.

Beside the backbone-based protocols, other approaches that rely on clustering techniques have been proposed to provide various solutions for the global service discovery problem in dynamic networks especially MANETs. A representative of the clustering approach is the Service Rings [57]. In Service Rings a number of clusters are formed. Each cluster (called a ring) of service providers is formed based on physical proximity and semantic proximity of the description of provided services. Every ring has its own Service Access Point (SAP), which is responsible for handling service registrations and service requests. Global discovery is possible by relying on such protocols. If a node's request is not satisfied by its local SAP, the request is forwarded to neighboring SAPs that are possibly capable of satisfying it. A similar approach is also adopted in [56] with the difference that the hierarchy of clusters is strictly dependent on a common service ontology. In [111], on the other hand, each cluster groups nodes with similar mobility patterns. In each cluster one of the nodes (called clusterhead) stays awake permanently and answers discovery requests. The rest of the nodes periodically wake up to provide the actual services and also to inform the clusterhead about their presence and services. The clusterheads are reelected periodically yo avoid draining a single node's battery.

The other solution to the global discovery problem, is to use *Distributed Hash Table* based techniques along with *location information*. Such approaches are described in [113] and [115]. The network topology is divided into geographical regions, where each region is responsible for a set of keys representing the services of interest. Each key is mapped to a region based on a hash-table like mapping scheme. A few elected nodes within each region are responsible for storing these keys, thus acting as directories. Global discovery is possible since a node requesting a service, uses the same hashing function and finds the directory location where its description is stored. Location information is also used in [122] for creating clusters of nodes based on physical proximity. Every cluster has a gateway, which is responsible of handling routing and discovery requests and for storing service descriptions from nodes located within its region. Inter-gateway request forwarding is also possible for global service discovery. This approach however differs from other backbone or cluster-based approaches in the sense that the gateways are elected or dismissed automatically based on location information and do not need to keep contact with each other.

Directory-less Architectures

In this type of architecture, the simplest solution to overcome the absence of service directories as an intermediate between service providers and clients is broadcast. This can be performed by both service providers and clients that broadcast service advertisements and requests respectively.

A basic problem in those non directory-based approaches is how to determine the frequency of service advertisements in order to reduce network load and avoid redundant transmissions. *Scheduling and prioritization* was one of the first techniques to deal with this problem. For example in [94], service providers periodically broadcast service advertisements to their one-hop neighbors. These advertisements contain services provided locally by the sending node and also services that the sending node has learned from its neighbors. Services that are about to undergo a state update are assigned a greater probability to be in the next broadcast. A similar approach is employed in [12] where a provider postpones advertising its services and backs-off for a fixed amount of time if it receives an advertisement that contains its own services. Very close to this concept is also the mechanism proposed in [68] where a provider listens to advertisements and when it is its turn to broadcast, it broadcasts only the service information that has not expired and has not been seen recently in previous advertisements.

Self-pruning broadcast techniques have been introduced to reduce the number of forward nodes in the network. In [134], the proposed approach is based on selecting a a small subset of nodes to form a forward node set to carry out a broadcast process. Each node upon receiving a broadcast packet, determines whether to forward the packet based on two neighborhood coverage conditions. These coverage conditions depend on neighbor connectivity and history of visited nodes.

Another way for lowering the network load consists in replacing broadcast by *muticasting*. In [44] service providers multicast their advertisements on a fixed multicast group and so do service requesters. In contrast to [94] where one-hop discovery is performed, in [44] the messages cover the whole network.

Nevertheless, covering the whole network using either broadcasting or even multicasting is very costly. This is why many approaches use various other techniques, such as advertisement range bounding in [17, 105], selective, probabilistic and intelligent forwarding in [33, 37, 91] along with the peer-to-peer (P2P) information caching, and the intermediate node responding to service requests in [88]. We will discuss these different approaches and protocols in the following part of this chapter.

Hybrid Architectures

Hybrid architectures represent a combination of both directory and directory-less architectures. In these architectures, service providers within the vicinity of one or more service directory will register their available services with the service directories. Yet, service providers must also be ready to respond to flooded service discovery requests from clients. When a client performs a unicast of a service discovery request to a directory, the directory responds with the information of the service provider if available. However, if there is no directory in the client's surroundings or if the directory does not have the

desired service, the client will simply broadcast the request to the whole network. In this case, service replies may come both from service providers and service directories.

Discussion Despite the multitude of publications on each of the service discovery architectures, researchers have not come into a general consensus on which architecture is better. Several factors may affect the general conclusion on which architecture is more suitable. For example, the nodes density, nodes mobility, service request frequency, etc. may play an important role in putting forward one architecture over the other. For our work, the centralized directory architecture is not a good solution for the volatile networks we are dealing with, since the service discovery process is mainly dependent upon the availability of the central directory, which represents a single point of failure, such as bottlenecks. A central directory hosted by a single node cannot be always accessible to all the nodes of the network, and such a node may not have enough resources to serve all the network nodes rising both scalability and global service discovery issues. Consequently, a directory-less architecture is the most suitable approach for the type of networks we are dealing with in our work. Keeping in mind that we try to limit the service provider role to stationary nodes (the infrastructures) that are capable of providing a large number of services and are capable to offer high computational and battery resources.

4.1.1.2 Service Discovery Middlewares and Protocols

There have been intensive research efforts in the field of service discovery protocols. Service Discovery Architectures like Jini [87], Salutation [19], UPnP [22] and SLP [39] have been developed over the past few years to efficiently discover infrastructure-based services from wired as well as wireless networks. Such type of networks are considered relatively stable, where the nodes can join and leave the network due to their mobility but such connections and disconnections occur at a slow rate. Typical examples of such networks are local networks in home and enterprise environments, where computers can be connected using wired LAN or wireless WiFi access points. Due to the few number of communication disruptions, service provisioning systems for such type of networks focus on discovering the newly connected nodes and the services they offer. Once discovered, invoking these services is not considered a challenge because a network connection is guaranteed between the service provider and the client. Such type of networks rely on a centralized approach to perform the discovery process, but this is not relevant for the type of networks we consider in our work.

Other service discovery protocols have been proposed to overcome the limitations of centralized mechanisms, such as the bottleneck problem and the single point of failure, by unifying on using decentralized architectures. Moreover, the dynamicity of the devices participating in the network have been taken into consideration in several research works. In the following section, we enumerate some selected discovery protocols for MANETs, and discuss their various characteristics and point out why they are suited or not for ICHNs.

The following service discovery protocols rely on one of three possible modes to provide clients with service information*: pull*, *push* and *hybrid* service discovery modes.

In the pull service discovery mode service providers (similarly service directories) do not distribute any advertisement of the services they offer, but clients on the other hand issue "service-searches" of the services they desire. The main problem this mode suffers from is the load imposed on the network by all the requests emitted by the clients. To avoid flooding the network with these service requests, such approaches rely on several solutions derived for lowering such a load. For example, some solutions propose limiting the time-to-live (TTL) value or bounding the number of hops traveled by the requests to control the message dissemination in the network. Other solutions propose utilize one of different forwarding mechanisms (selective, probabilistic or intelligent) or even like in [65] apply geographical limitations over the propagation of the messages beyond a specific geographical area instead of forwarding the requests to all the nodes in the network.

On the contrary, in the push mode clients do not generate any service-searches, but service providers advertise their services on discrete time intervals. Similar to the service-searches messages, these advertisements are subjected to several limitations, such as limiting their forwarding to bound ranges instead of flooding the whole network. Moreover, the frequency of emitting advertisements is define by service providers according to the level of dynamism of the environment we are dealing with, for instance, if the mobility of the nodes in the network is high or the percentage of failures in message transmissions is high, then service providers should generate service advertisements more frequently. Otherwise, if the network we are dealing with is prone to congestion, then the rate of emitting advertisements should be decreased.

This mode merges both push and pull modes where service providers and clients may issue advertisements and service-searches respectively. For example, service providers may proactively advertise their services to service directories, but clients may issue service-searches to service directories only reactively. The frequency of emitting these messages greatly depends on the properties of the nodes forming the network. As explained in [79], when all the nodes in the network are "greedy", a strategy is implemented where all service providers may advertise services to all nodes and all clients may send service-searches in order to discover the desired services. On the other hand, if the nodes contributing in the network are "conservative", then a different strategy is implemented where service providers may reduce the amount of generated advertisements to cover a random set of nodes and clients may also send service-searches only to a random set of nodes, since nodes are willing to contribute in disseminating both types of messages. The third and more complex mechanism discussed in the hybrid service discovery mode relies on a more intelligent approach to avoid forwarding requests to the same set of neighbors, where a node memorizes the previously contacted nodes and increases gradually the size of the targeted set of nodes till the service is discovered.

Although in MANETs most of the research work done on service provisioning also focused on the service discovery process, they proposed discovery protocols tightly coupled to routing protocols so as to minimize the cost of communication. In these protocols, the requests and the responses are often integrated into routing messages by piggybacking. Proposals described in [139, 125, 61] are examples of such protocols. In [125] for example, the authors argue that service discovery can be greatly enhanced in terms of efficiency, regarding service discoverability and energy consumption, by piggybacking service information in routing layer messages. Thus, a device requesting a service in ad-

dition to discovering that service, it is simultaneously informed of the route to the service provider. In their assumption, they extended the Zone Routing Protocol (ZRP) [40] in order to encapsulate service information in its routing messages. However, these protocols assume that communication between two mobile devices are possible only if they are present simultaneously in the network and that there exists an end-to-end path between them.

DEAPspace [94], a service discovery protocol for single-hop ad hoc networks. It relies on a pure push-based mechanism to perform the discovery process. Servers periodically broadcast service advertisements to their one-hop neighbors. A service in DEAPspace is described using a compact predefined format containing its name, address, inputs, outputs, properties, and time-to-live. Each node maintains a list of its local and discovered service descriptions. In DEAPspace, every device maintains a view of all the services present in the network, but this is accomplished without taking into consideration the load imposed on the network by broadcasting.

Unlike DeapSpace, **Konark** [44] is a service discovery and invocation protocol for multi-hop ad hoc networks that was designed in the aim of avoiding the broadcast storm by relying on a mixed push-pull mechanism for discovery. A service description is based on the Web Service Description Language (WSDL) [129]. Each node maintains a local service registry that stores service descriptions in a tree based structure. Konark relies on a hierarchical classification of services, thus service providers multicast their advertisements on a fixed multicast group. When a client needs a service, it multicasts a search query. If the search query is received by a node that has the matching services in its registry, the latter advertises a message concerning the desired service. By that, Konark intends to expand the limited scope of DEAPspace that is caused by the one-hop discovery process by multicasting the messages to cover the whole network.

DEAPSpace and Konark are examples of discovery protocols developed at the application level, where discovery is performed on top of a routing protocol, and no assumption is made on this one. DEAPspace allows discovering and invoking services in the immediate neighborhood (i.e., at one hop). Konark middleware has similar objectives as DEAPspace but considers multi-hop discovery and invocations. Moreover, each service provider in the network is equipped with a micro-HTTP server that handles client invocations. In turn, invocation requests and responses are based on SOAP over HTTP. However, Konark makes a stronger hypotheses on the lower layer as it assumes that a route can be used between the client and the service provider for invocation.

Despite the attempts to avoid the excessive load on the network, but disseminating messages using either broadcasting or even multicasting techniques is very costly. The **Group-based Service Discovery Protocol (GSD)** [17], defined by Chakraborty et al., introduces an additional technique beside multicasting to reduce the effect on the network load. GSD is a service discovery protocol for pervasive and ad hoc environments that bounds the range of the general advertisements, thus an advertisement message will be dropped after a predefined number of hops. A service in GSD is semantically described using the OWL [128] ontology language, which enables the hierarchical grouping of services. However, GSD expands the dissemination of the available services by relying on peer-to-peer caching and semantic matching of services, allowing nodes to merge services and re-advertise them. Although this approach solves the broadcast storm problem and reduces the load significantly in the network, but it makes some assumptions on the

nature of the service interfaces that can be supported in the network due to the need of the predefined ontology to group similar services. In GSD [17], beside the service discovery protocol, Chakraborty et al. also proposed a service invocation protocol GSR-S [16], that uses the discovery routes created by GSD to support invocation. It specifically uses the route created by advertisements or the route created by the discovery request, and assumes that the links are reversible. Moreover, it enables session maintenance, where a client can initiate a session with a specific service provider or with the service instance independently of the service provider.

The **Pervasive Discovery Protocol (PDP)** [12] is a discovery protocol for ad hoc networks with limited device resources. A node in PDP contains a service cache containing a list of known services. A service is described using a description language called Generic Service Description Language (GSDL) based on XML, it enables the description of hierarchical relationships among services. A service provider broadcasts the descriptions of its services on demand only. Consequently, all the devices in the broadcast range store these service descriptions in their cache. When a service provider receives a discovery request from a client, it waits before forwarding the response for a calculated amount of time to listen if a similar response is being advertised by another provider. A client can send a service request to one or more providers providing the same type of services. Finally, if a service provider wants to switch off, it sends a deregister message to all the other devices. The invocation process implemented with the PDP is performed using the SOAP requests and responses.

In [91], the authors propose **Service*** a pure push-based service advertising scheme that relies on intelligent forwarding to spread service advertisements without forcing any ontology or requiring a specific service interface. In this technique, the costly broadcast is replaced by unicasts. Thus, Service* tries to reduce the message overhead in the network by eliminating network-wide packet dissemination and by optimally placing the services in a dynamically selected subset of the available nodes, called brokers. By the usage of brokers, a service provider performs a unicast of service advertisements toward these nodes through which all the 2-hop neighbors can be reached. As a result, every service provider is continuously monitoring its 2-hop neighborhood and every node in the network can be reached while avoiding the duplicate forwarding.

In [88], the authors aim at both reducing the load imposed by the service discovery and advertisements and increasing the number of discovered services by clients in the network. The authors state that relying only on the cache mechanisms to disseminate the services in the network has the advantage of minimizing service discovery request flooding. However, it has the potential drawback of lowering the number of discoverable services. Therefore, in their work they base on the hypothesis that intermediate nodes may have been informed about the existence of some services either by receiving and forwarding service advertisements or because they themselves have requested these services in the past. As a consequence, they propose allowing intermediate nodes to respond to service requests. However, in order not to decrease the number of discovered services the authors propose that intermediate nodes must be informed of all the services matching the issued requests. This can be performed by forcing all the intermediate nodes to update their service lists whenever they encounter any response to a client's request. Thus, all intermediate nodes will be up to date and will be able to reply with all the matching services they became aware of by informing each other in previous requests.

Discussion There exists a large number of service discovery protocols for MANETs, we selected a limited number of these protocols to spot the light over the different approaches present in the literature. Many surveys are proposed in the literature [114, 43, 126] that try to classify these systems according to various criteria. Despite the fact that these protocols consider the dynamicity of the targeted networks, they mainly assume the availability of a fully-connected underlying network, which can support an end-to-end path between any two nodes in the network. Obviously, such assumptions cannot be applied over intermittently connected networks, where the volatility of the devices, the short communication range of wireless interfaces, accompanied with the radio interference and the free movement of people results low density networks with no guaranties on the end-to-end paths on the reliability of communications. Consequently, the provision of application services with this kind of opportunistic and asynchronous communications needs specific mechanisms to deal with all the constraints these networks suffer from. Nevertheless, some ideas, such as those presented in [91] and [88], can be considered interesting for service provisioning in ICHN, as they propose allowing intermediate nodes to respond to service requests. Thus, overcoming the absence of end-to-end path constraint of ICHN networks. Indeed, clients are not obliged to contact service providers or registries to invoke a specific service, as any intermediary node can resend advertisements or even responses it has already received.

4.1.2 Service Invocation in MANETs

In MANETs, the service invocation process has not been considered as a problem because the network is considered as connected thanks to the underlying routing protocol. So, traditional end-to-end service invocation can be used directly.

4.2 Service Provisioning Systems in ICMANETs

Service provisioning in ICMANETs using opportunistic communications is an emerging computing paradigm that has been recently qualified as opportunistic computing [20]. This paradigm introduces new issues regarding both the service discovery process and the service invocations process. The routing protocols must be suited to both discover and deliver pervasive services.

4.2.1 Service Discovery in ICMANETs

A limited number of studies have been introduced tackling the service discovery problem in ICMANETs. Conti and Kumar [21, 20] focus on service provisioning in opportunistic networks that are solely composed of mobile devices. They target the networks relying on the social interactions between the mobile devices that act as both clients and service providers. In their work, the authors focus on building a services platform with its key element the discovery of the services in a mobile and pervasive environment. Their focus was on acquiring the knowledge of when and where services are available. Beyond discovering the services in the network, their work aimed at predicting the service availability taking into consideration when and for how long a node can meet the device

providing a given service and its resources availability (e.g., the amount of energy or the traffic load on that node). By that, they did not introduce an invocation mechanism but relied on direct communication between the client and service provider.

Another service discovery approach introduced in [81], where the discovery mechanism is implemented over a content-based communication facilities exploiting the publish/subscribe paradigm. Instead of relying on a global service directory, the authors rely on a peer-to-peer approach in which each provider proactively advertises its services by publishing them in the network and the clients are informed of the existence of the services they need through subscriptions.

In [66], Le Sommer et al. present a proxy-based model for an enhanced provision of application services in opportunistic networks. This model relies on a taxonomy of application services and a content-based management of services messages. It allows service providers to specify which services can be delivered by a proxy or a set of proxies and to select which mobile hosts should act as proxies either implicitly using conditional rules or explicitly by specifying the addresses of the mobile hosts. In this model, the services are provided fixed infostations in limited geographic areas. Similar to [81], this model is based on the "store, carry and forward" principle and on a content-based management of service messages.

In [102], the authors introduce the service delivery platform called SCAMPI (Service platform for social-aware mobile and pervasive computing). The SCAMPI architecture combines the distributed task execution with social and context-aware opportunistic networking to enable opportunistic computing in pervasive networks characterized by a rich set of resources. The key elements of the architecture include leveraging human social behavior for efficient opportunistic interaction between a variety of sensors, personal communication devices and resources embedded in the local environment. The SCAMPI architecture abstracts resources as service components following a service-oriented model. This enables composing rich applications that utilize a collection of service components available in the environment.

A middleware for location-based service discovery and invocation in ICMANETs have been proposed in [63]. The protocol OLFServ, proposed in [63], is designed to perform a geographically-controlled broadcast of service advertisements and service discovery and invocation requests. It enables service providers to specify in their advertisements their location and the geographical area where they can be discovered and invoked. Moreover, it allows clients to define their location, the area where their messages must be disseminated, the location of the provider they want to invoke, as well as their moving direction and speed if they know them. This protocol implements self-pruning heuristics allowing mobile devices to decide whether they efficiently contribute in the delivery of the messages they receive from their neighbors.

Discussion A lot of research efforts have been focused on designing service discovery approaches in wired networks and connected (stable and dynamic) MANETs. But these approaches cannot be simply deployed in ICMANETs. In such types of networks no device is considered stable enough, or accessible permanently, to act as a service registry. Each mobile client should therefore be responsible for maintaining its own perception of the services offered in the network. Yet, to our knowledge, only limited efforts have been

introduced to provide new solutions for service discovery in ICMANETs.

4.2.2 Service Invocation in ICMANETs

Similar to service invocation in MANETs, service invocation in ICMANETs has been also tackled as a routing problem. Indeed, this challenge can be represented as a problem of routing messages (request messages from a client to a service provider or response messages from a service provider to a client) between two nodes in an ICMANET. Besides the various routing approaches already presented in Chapter 3 that can be used as solutions for the service invocation challenge in ICMANETs, we present some of the protocols that were designed to specifically target service invocation in ICMANETs.

In the content-based communication model nodes generally relay messages according to the content of these messages, making the flow of information interest driven rather than destination driven [14, 23, 54]. As shown in [81], Mahéo and Said use the publish/subscribe facility for implementing content-based invocations. In their middleware, the authors employ the "store, carry and forward" approach for the network-wide opportunistic dissemination of messages. The service invocation process is implemented on top of these communication features. The client formulates a request that includes a reduced version of the service description and publishes it in the network. Providers subscribe for any request matching the reduced version of the descriptors of the services they provide. The provider's response is transferred back to the client thanks to the publish/subscribe mechanism, where mobile clients can establish some correlation between the invocation requests and the responses.

Another protocol targeting service provision is ICMANET is OLFServ [64], a location-aware protocol designed in order to support both service discovery and service invocation in ICMANTs. Based on the location data collected from the wireless interface and/or GPS receiver of the device, OLFServ makes it possible to perform an efficient and geographically-based broadcast of both service advertisements and service discovery requests. Beside the location data, this protocol implements several self-pruning heuristics aiming to efficiently control the dissemination of service advertisements and service discovery requests, as well as to perform a geographic and source-based routing allowing cost effective delivery of service invocation requests and responses. Consequently, this protocol exploits the device mobility in order to achieve a network wide message dissemination, and allows nomadic people to have access to services offered by infostations even if they are not in the area covered by these devices. Nevertheless, this model requires to guide the message propagation using contextual information, and especially location information.

Discussion In recent years, a wide range of approaches have dealt with the routing challenge in ICMANETs. Nevertheless, almost non of these protocols are designed to target ICHNs. Indeed, the majority of these protocols represent the network as formed solely from mobile nodes (both clients and service providers are mobile devices). As a first insight, it may seam logical to consider that any routing protocol initially designed for ICMANETs should work in ICHNs, since ICHNs are a special case of ICMANETs. However, this is not straight forward. In our work, we try to show the importance of

utilizing an invocation protocol specifically designed to target ICHNs and take into consideration the various stationary service providers in the network. We show that considering these facts in the designing phase will have a noticeable effect on the generated overhead of the protocol, its delivery ratio and even on the over all performance of the protocol.

4.3 Handover for Service Provisioning

Since we are dealing with ICHNs in our work, we should take into consideration the importance of access continuity during the mobility of the client in the network. Our aim is to support access continuity of mobile clients to services while moving in the network and switching from one infostation to another. Indeed, the cooperation between wireless infrastructures and opportunistic networks has been recently investigated in order to enhance the content delivery to mobile clients, to support access continuity and to reduce the load of the infrastructure. In the following section we will introduce several related works that have dealt with such issues.

The Handover in wireless networks can be classified into two major categories: *horizontal* (between different parts of the same network) and *vertical* (between different types of networks). Horizontal handovers usually occur in homogeneous networks. A homogeneous network is a network that relies on the same communication protocol between the different nodes forming it (all nodes are equipped with the same type of wireless interface). In a homogeneous network, horizontal handovers are typically required when an infostation[1] becomes unavailable due to the movement of the mobile device, i.e., when the device move away from the vicinity of the infostation. Horizontal handover can also be present when moving between two networks that use the same network technology and interface.

A vertical handover, on the other hand, is mainly needed in a heterogeneous network in which users can move between different access networks. It allows the switching of the ongoing network connection from one wireless interface to another (e.g., handover from an 802.11n network into a GPRS network). Generally, the need for vertical handovers can be initiated for convenience rather than connectivity reasons (e.g., according to user choice for a particular service).

Both horizontal and vertical handovers can be qualified as *hard* or *soft* handovers. With a hard handover mechanism, a mobile device can be connected with only one infostation at the same time. It is referred to as a *break before make* handover. While a soft handover mechanism allows to keep two or more connections with different infostations at the same time, it is referred to *make before break* handover.

Between vertical and horizontal handovers, the horizontal handover mechanism is the apparent solution that suits the type of network we are concerned with in our work. Indeed, horizontal handovers are dedicated for ensuring a seamless switching from one access point to another or from one infostation to another in the same network.

For example, in [108], the authors propose a layer 2 handover mechanism for the IEEE 802.11 that is able to eliminate the scanning delay and reduce the total handover delay.

[1] We use the term "infostation" to denote a fixed service provider.

In general, the layer 2 handover process occurs between access points within the same IP subnet. In this work, the authors eliminate the scanning phase performed by the mobile devices when moving between different access points in the same network. They do so by forcing the access point to forward the information used by the access point to build its neighbors table. Thus the mobile node uses this neighbor table when a handover is needed to connect directly to the next possible access point without the scanning phase.

In [51], the authors aim at disseminating content to mobile nodes while meeting guaranteed delays and minimizing the load on the wireless/3G infrastructure. This framework consists of a control system which pushes content to mobile nodes and keeps track of its opportunistic dissemination. The framework relies on a close-loop controller to decide when to push new copies of the content through the infostations and to which set of nodes these messages should be forwarded to ensure a smooth and effective dissemination using epidemic routing. Thus, Push-and-Track favors the opportunistic ad hoc communication over the direct 3G connection between the smartphones and the 3G infrastructure whenever possible. Although the global infrastructure load is reduced by relying on this framework, relying on the epidemic routing protocol and requiring acknowledgments to keep the loop closed may significantly increase the network overhead.

The work in [73] relies on DTN to offload multiple mobile data traffic from overloaded cellular networks to high capacity and free device-to-device networks. Mobile nodes that are used to offload traffic are called "helpers". In their work, the authors take into consideration the different delay sensitivities and sizes of the offloaded data and that the offloading helpers' storages are limited in size. They propose three algorithms depending on the level of complexity present in the offloading scenario and on the life times of the offloaded data. Algorithms with higher complexity are used in cases where the data is not delay sensitive, where the algorithm with the lowest complexity is limited to offloading scenarios with similar contact rates and content sizes. In their system, the authors assume that all the helpers cooperate in the offloading scheme. However, in practice, helpers may act strategically to offload data for others.

Discussion Although several handover mechanisms already exist in the literature that target such a problem, but such approaches does not match the requirements of the networks we are dealing with. While present handover mechanisms require an end-to-end path between the mobile device and the infostation, in our case the handover mechanism should be able to choose one infostation over the other even if the mobile device is not found in its coverage area or no end-to-end path exists between the mobile device and the infostation. Indeed, the handover mechanism should take into consideration that a device can connect to an infostation even if it is not located in its vicinity with the help of the opportunistic communications of the mobile devices in the network.

4.4 Conclusion

In this chapter, we have presented the evolution in the proposed approaches that deal with both service discovery and invocation. Spite of the fact that hundreds of protocols are available for service provisioning in MANETs, but they are designed based on the

assumption that end-to-end connectivity is always available, and that nodes participating in the network are reliable enough. Such conditions are hardly present in ICMANETs. For that, new approaches have been proposed for service provisioning in ICMANETs. We also introduced some works that target service provisioning in ICMANETs besides those introduced before in Chapter 3. The importance of designing a service provisioning protocol specifically targeting ICHN to be discussed in the following chapters. Finally, we gave an overview of the available approaches that deal with access continuity to enhance the content delivery to mobile clients in the network.

Part III

Contributions

Overview of the Contribution

The goal of the proposed middleware oriented framework is to provide network-wide service provision in ICHNs. Targeting clustered ICHNs, in this thesis we focus on two main issues to efficiently provide nomadic people with pervasive services in ICHNs, namely the discovery and invocation of services using opportunistic communications. In addition, due to the hybrid nature of the network structure, we propose a handover mechanism to offer a service access continuity to the mobile clients.

In our solutions, we tend to show the importance of designing a specialized protocol for ICHN over using a general purpose protocol to perform service provision. The protocol should be able to handle the communication constraints in the mobile part of the ICHN, where the device density and mobility produce fragmented network topologies. Moreover, the protocol should adapt to the various communication patterns relative to each phase of the provision process.

For instance, in the discovery phase, it is important to take into consideration the fact that services are solely provided by fixed infostations and clients run on mobile devices. Our objective here is to inform all potential clients of the provided services in the network and make them aware of any modifications of the state of these services as quickly as possible.

As for the invocation phase, the problem can be viewed as a unicast communication pattern between a mobile client and the infostation providing the service. However, since we target clustered ICHNs, we exploit the presence of fixed and inter-connected devices in the network in order to improve the service delivery. In this context, sending an invocation request can be viewed as an anycast problem.L

The physical position of a client throughout the invocation process. This is a result of the free mobility of clients coupled with the delay subjected on the message dissemination in ICHNs. The presence of connected infostations in the network can be also exploited to optimize access continuity for clients to provided services. We consider introducing a handover mechanism to improve content delivery in the network.

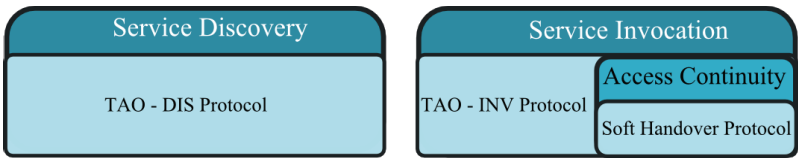


Figure 4.2: Overview of the Middleware Oriented Framework

In the following chapters, we present our proposal for a novel platform, called TAO (Time-Aware Opportunistic platform), to perform service provision in ICHNs. TAO is a platform that provides an API for service discovery and invocation in ICHN. It is composed of two protocols: TAO-DIS and TAO-INV and implements a handover mechanism. Figure 4.2 gives an overview of the platform we are proposing. TAO-DIS protocol is introduced as a solution for the service discovery challenge in ICHN. Similarly, TAO-INV protocol is the solution we propose for the service invocation challenge in ICHN. The access continuity challenge is tackled by the soft handover protocol that is viewed as an optimization of TAO-INV.

5

TAO-DIS: a Protocol for Service Discovery in ICHNs

In this chapter, we present TAO-DIS protocol, the part of the platform that supports the service discovery phase. Our main objective here is to provide an automatic, low-overhead and light service discovery mechanism. TAO-DIS achieves this objective by considering a special representation of the service descriptors and assembling them in what we call a *Service Guide*. A selective forwarding mechanism is applied to the resulting service guides couple with the peer-to-peer caching concept. This chapter gives a detailed representation of TAO-DIS and the mechanisms it relies on.

5.1 Push *vs* Pull

By definition, service discovery enables network devices to become aware of the availability and capability of peers in the network. Service providers describe and advertise their capabilities for clients to discover and match their needs. Achieving network-wide service discovery while handling the communication constraints inherent in the intermittently connected mobile environments is a complex problem.

The main approaches considered in performing service discovery are the push-based and pull-based mechanisms. Several already present discovery protocols have relied on one or both (merged) approaches. However, the exchange of messages in intermittently connected networks is not instantaneous as in connected networks. A push-based approach, that involves only one-way communication, is likely to be more efficient than a pull-based approach in an intermittently connected network. Indeed, in a pull-based approach, clients actively initiate a dissemination of requests for the services they seek and providers reply by unicast messages containing the list of proposed services toward the clients. This incurs a two-way communication that suffers more from the network disruptions. Actually, the delay subjected on the propagation of messages (both requests and responses) is not negligible. For that, it is inconvenient, most of the time, for users to wait till the discovery process terminates to choose the desired services and invoke on or more of them.

However, with a push-based approach, the delay is greatly reduced in comparison to the pull-based approach. Since clients do not generate any requests to receive the list of proposed services, the delay induced from the propagation of the request messages is eliminated. Keeping in mind that, in our work, we focus on city center and similar sce-

narios. In such scenarios, the number of infostations is noticeably lower than that of the mobile nodes. Besides, each infostation provides a limited number of services thus the number of service requests is higher than the number of offered services. Nevertheless, the most important drawback we should be aiming to overcome is the number and size of messages generated by service providers to advertise their services. Indeed, in this approach service providers generate advertisements and forward them to all the users in the network even if some users might not interested in the proposed services. Thus, the overhead resulting from this approach can be considerably higher than that from the pull-based mechanism.

To this purpose, we tend to perform an efficient, automatic and fast service discovery process in ICHNs. In the following section, we detail the discovery protocol we propose named TAO-DIS.

5.2 Overview of TAO-DIS

The purpose of TAO-DIS protocol is to perform an automatic, low overhead and light service discovery mechanism. The protocol works in a typical scenario where a client and a service provider (infostation) are considered distant with no end-to-end connections. The same protocol remains valid in scenarios where clients and providers are in proximity. The discovery protocol consists of all the provision steps needed before an invocation, which are:

- Description and advertisement that are performed by the service provider.
- Collection and selection that are performed by the client.

In practice, service providers represented by the infostations create the descriptors of the services they are willing to support. These descriptors provide all the necessary information needed by potential clients to match their interest with the provided services. Mobile nodes participate in the dissemination process of information by relying on the "store, carry and forward" principle. However, the number of messages and amount of data exchanged should be taken into account for its significant effect on the performance of such networks.

In an ICHN, infostations may be gathered in a form of administrative groups. For example, a local authority may manage a set of infostations distributed in a city center, each of them providing information services related to a historical monument. As a result, these infostations form clusters. Within a cluster, infostations are connected together so they know the services offered by the others. On the contrary, infostations are unaware of infostations belonging to other clusters, and hence of the services they provide. Therefore, identical services provided by two infostations located in distinct clusters will be considered as different.

Once a client has received some service descriptors, it can examine their functional and non-functional parts in order to select the service best suited for its needs, for further invocation. Using a form of full epidemic as a means to disseminate service descriptors is the fastest way to reach the whole network. In such an epidemic, whenever a device encounters another device, it transmits a copy of its descriptors (if the encountered device

does not own them yet). But this process is known to be very costly in terms of bandwidth and cache usage, a fortiori because a full dissemination is necessary at each addition, deletion or modification of a service descriptor. So TAO-DIS implements service discovery in a form of full epidemics that includes optimizations with two objectives:

- Ensure that all service clients are informed as quickly as possible of every change in the offer of services by service providers (service creation, deletion or modification).
- Reduce to the minimum the number of messages and the amount of data exchanged between devices in this discovery process.

5.3 TAO-DIS Protocol

5.3.1 Service Descriptors

The first step in the service discovery process is the description of the services by the service provider, using all the information needed by the potential clients. Thus, the service provider envelops the description of the functional and non-functional properties along with the provider's properties in a single message called the service descriptor. Therefore, a descriptor is essential since it contains all the information needed by the client to identify the provided services.

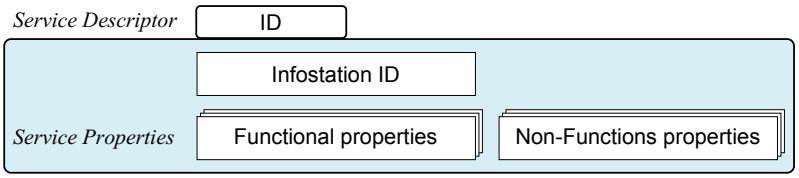


Figure 5.1: Components of a service descriptor message

As depicted in Figure 5.1, the main components of a service descriptor are: (1) the infostation's identity, (2) a set of the service's functional capabilities and (3) a set of the service's non-functional details. Each service descriptor is identified with an unique ID.

The service description aims to list the functional capabilities and the non-functional details. It is the job of the service creator (programmer) to define these functional and non-functional properties. The functional properties list the signatures of the different functions offered by the service. These properties form the service interface that a client needs in order to understand the functional aspect and the tasks performed by the service. Figure 5.2 shows a functional description of an example service "S1" offering the capability "add" that takes two numbers and returns their sum, written in JSON (JavaScript Object Notation).

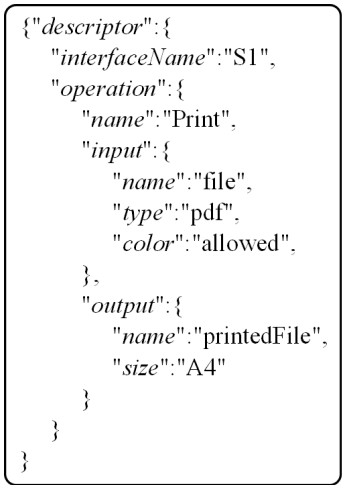


Figure 5.2: Example of functional service properties

The non-functional properties give details about the service provision. Extra details enhance the service description with non-functional properties that are unrelated to the tasks performed by the service but they represent the policy aspect (e.g., author, version, cost, quality, etc.). These non-functional properties can be decisive when a user chooses between many providers of the same functionality, such as the security or the quality of service. Figure 5.3 shows an example of non-functional service properties written in JSON.

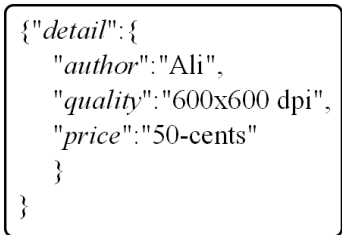


Figure 5.3: Example of non-functional service properties

We use for this part a list of key-value pairs. There is no restriction on the language of the descriptors. For example, WSDL could be chosen for an important expressiveness in the description and the selection of services. In our implementation, we limited the types of the functions of the services by using JSON for formatting service descriptors.

For each descriptor, a unique identifier should be generated. However, the presence of several clusters in the network rises some problems in ensuring the uniqueness of these identifiers. Infostations belonging to the same cluster have the ability to communicate with each other and thus ensure unique identifiers to provided services. On the contrary, this cannot be accomplished with other infostations that belong to different clusters as no direct communication is possible.

To overcome this problem, we introduce the *hash key* component to play the role of a unique identifier of each service. The hash key is the result from applying the hash function to the JSON descriptor. It can be generated locally even if there is no prior communication between the different infostations in the network. Thus, the hash key generation process is defined by:

$$HK = \mathcal{H}(Functional.Content + NonFunctional.Content + InfostationID) \quad (5.1)$$

Where $\mathcal{H}$ is the *hash function* used to create the hash key of the service, and the "+" operation is the string concatenation operation. For this implementation of our protocol we utilize the SHA-1 hash function [90].

5.3.2 Service Guide

After the presentation of the different components of the service descriptor, we focus on the dissemination of the service advertisements. Generating an advertisement for each provided service or for each update of the service state is not an efficient approach as it imposes a high network load. Since our aim is to reduce the resulting overhead from the push-based approach, we introduce the concept of a "*Service Guide*". The descriptors of services provided by the infostations belonging to a cluster x are grouped in a so called Service Guide SG_x. The overall objective of discovery is that a unique SG per cluster is eventually present on every mobile device. Actually, the nature of the ICHNs makes that each mobile device builds its own set of SGs (one per cluster) and tends to opportunistically synchronize them the other devices. SGs are originally created and advertised by infostations. Furthermore, on an infostation, every addition, modification or deletion of a service will result in the creation of a new version of an SG that will be advertised.

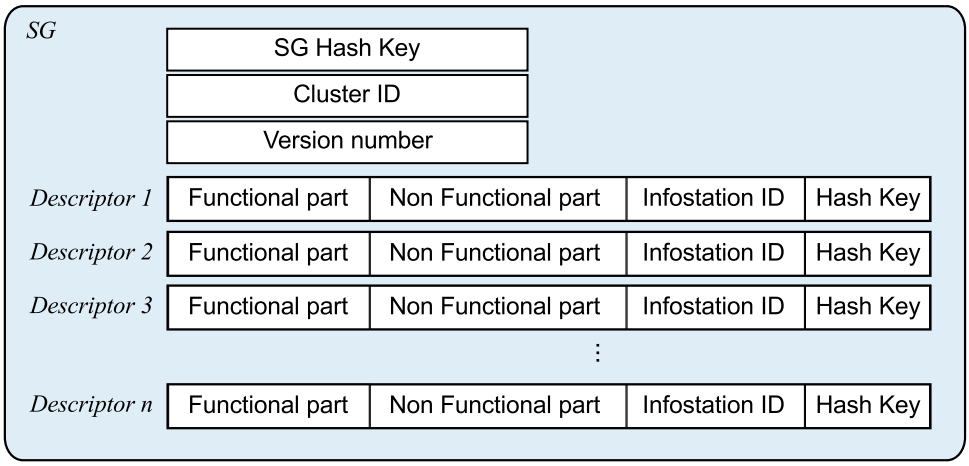


Figure 5.4: Components of a service guide

As depicted in Figure 5.4, apart from the catalog of service descriptors, an SG contains the cluster identifier and a version number. In addition, similarly to a descriptor, an SG

is identified by a unique hash key. The cluster identifier (Cluster ID) is used to relate this SG to a specific cluster for further comparisons during the dissemination process of the SGs in the network. The *Service Guide hash key* results from applying the hash function to the concatenation of the included descriptor identifiers. Then, the hash key of the SG is defined by:

$$HK_{SG} = \mathcal{H}(\mathrm{C}_{i=1}^{i=n} HK_{s_i}) \tag{5.2}$$

Where C is the string concatenation operation and HK_{s_i} represent the different hash keys of the various service entries in the SG.

In a given cluster *x*, infostations are connected together, so they cooperate in order that they issue consistent instances of $SG_{\mathbf{x}}$ (i.e., successive versions of $SG_{\mathbf{x}}$). Indeed, synchronization among the infostations of the same cluster is required especially in cases where two or more of these infostations might simultaneously introduce some modifications on the provided services.

5.3.3 Node Cache

Since the resources available in the mobile devices are considered scarce and limited, it is important to point out what kind of information TAO-DIS requires to be registered in the cache of these devices (both relays and clients). On the other hand, in the aim of reducing the amount of transmitted data, devices should store in their cache a list of *old service guide* (OSG). For each cluster *x*, a device stores in its cache a list OSG_x of several older versions (the *k* most recent ones, with *k* arbitrarily fixed) of SG_x it already knows. An OSG list contains only identifiers of descriptors. Whenever a device should send a new version of SG_x to its neighbors, it actually sends only the difference between an old version present in its OSG_x list and the new version. Under the assumption that the neighbor's version of the SG_x is already found in its found in its OSG_x list.

The number of older versions (i.e., *k*) that can be stored in an OSG_x list is directly related to the amount of resources each device is willing or capable of providing. Nevertheless, if a device is not capable of allocating a part of its resources for the OSG_x list, this will not severely affect the performance of the discovery mechanism. In fact, it is only necessary to keep the most recent encountered version of the SGs to perform a successful discovery mechanism with TAO-DIS and older versions can be simply dropped. However, each device should be capable of registering in its cache at least one SG for each cluster where the size of a SG is of the order of few kilobytes.

5.3.4 Dissemination of the Service Guides

TAO-DIS relies on a push-based mechanism to disseminate the SGs over the mobile users in the network. In general, at the creation of a new version of an SG the infostations present in the network forward this new version toward their one-hop neighbors. The mobile devices possessing these SGs have the responsibility of forwarding the new version of the SGs (or the difference between outdated versions and the current version of the SGs) toward other mobile devices in the network. The forwarding decisions are taken in a local manner inside the mobile device itself, based on the local knowledge of the state of its direct neighbors.

The dissemination of the SGs is ensured through a gossiping phase that takes place between one-hop neighbors, when a mobile device encounters an infostation or another mobile device. We assume that an underlying protocol covers the neighboring discovery, that issues an event destined to TAO-DIS when a new neighbor appears in the one-hop neighborhood. There is no difference of behavior between the infostations and the mobile devices regarding the gossiping phase. Nevertheless, the infostations are the only devices that generate new versions of SGs.

Each device stores in its cache the most recent version of an SG it knows, for each cluster. When a device encounters a neighbor, the first operation performed is sending an "*offer message*". This offer message is used by the device to inform its direct neighbors about the various SGs found in its cache. It contains, for each cluster, the hash key of the SG the device holds in its cache, the cluster ID of the relative SG and the version of the SG. As depicted in Figure 5.5, the offer message is formed of several entries, where the maximum number of these entries is limited to the number of clusters found in the network.

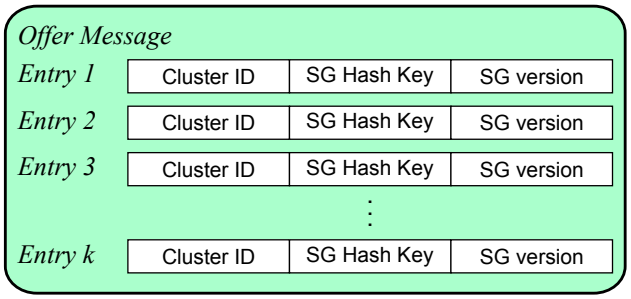


Figure 5.5: Template of an offer message

Upon reception of an offer message, a device compares this offer with what it knows, that is, for each cluster x, its own SG_x and the list OSG_x. If a newer version of SG_x is found locally, the device automatically (neighbor node does not ask for any updates) generates an update message containing the difference between the two versions (i.e., old version found in neighbor's cache and new version found locally) of the SG_x. This difference is computed locally by comparing its SG_x with the information retrieved from the OSG_x list. As a result, a list of descriptors entries to add or delete (a modification is implemented as a deletion plus an addition), is sent to the neighbor, or the device simply keeps ready to receive a similar update from its neighbor (symmetrically, the neighbor should decide to send it). Hence, most often, only differences between SGs are transmitted in update messages. It is only when no older versions of an SG is known by a device that it will receive an entire SG in an update message.

Let us consider a simple scenario to explain in details how such a process occurs (Figure 5.6). Assuming there are two clusters of infostations (respectively A and B) which

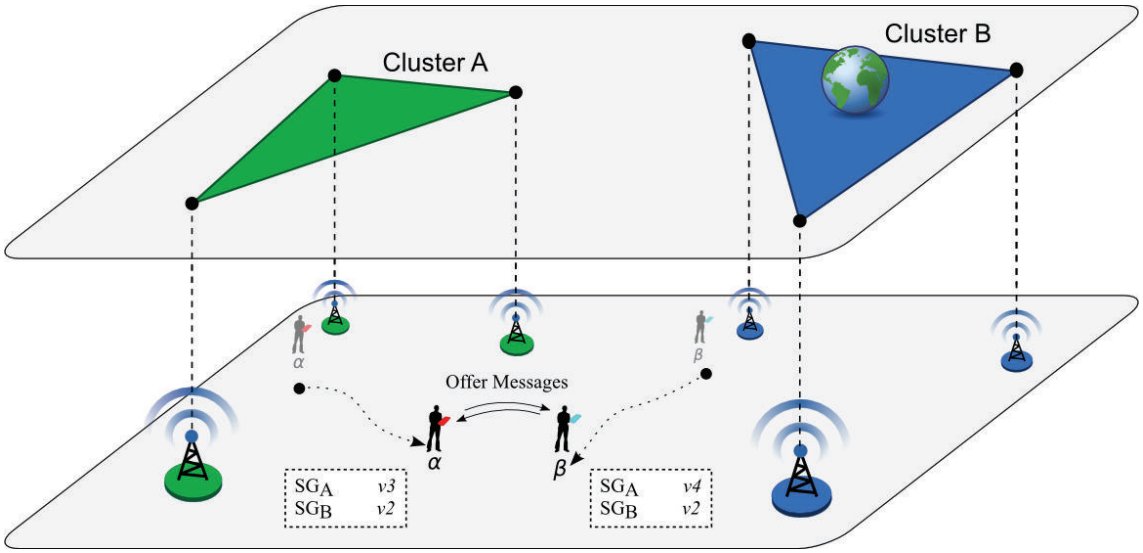


Figure 5.6: Example of SGs dissemination between mobile nodes in an ICHN

have advertised several versions of SGs (4 versions for cluster A and 2 for cluster B). Let α and β be two devices roaming the network. α has in its cache $v3$ of SG_A and $v2$ of SG_B, and in the OSG lists it has knowledge about $v1$ of both SG_A and SG_B. β, on the other hand, has in its cache $v4$ of SG_A and $v2$ of SG_B, and in its OSG lists knowledge about $v1$, $v2$ and $v3$ of SG_A and $v1$ of SG_B.

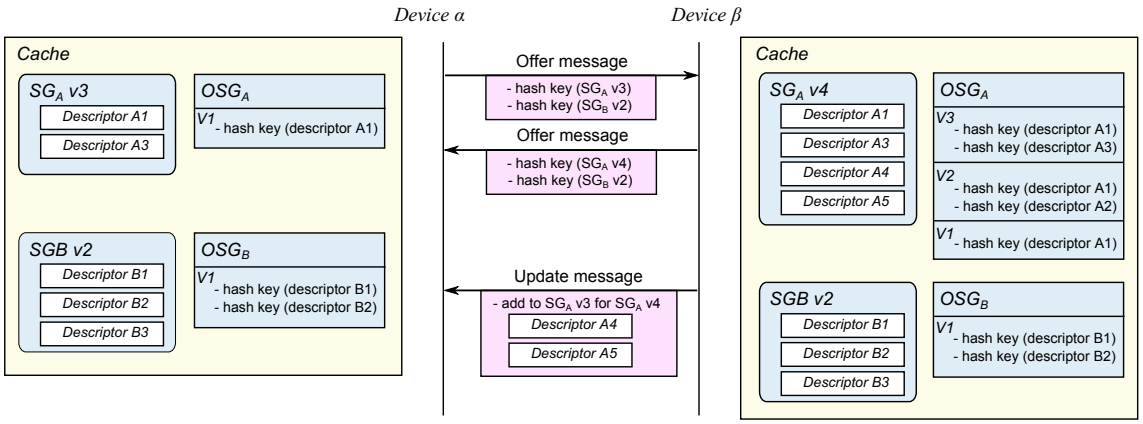


Figure 5.7: Example of a gossiping phase between two mobile devices

Figure 5.7 illustrates a typical gossiping phase between α and β. In the figure, the parallel reactions to the discovery interleave: one sees that device β sends an offer message to device α almost at the same time it received an offer. But this is only a possible behavior, as β may discover α a bit later, and would refrain from sending its offer, knowing that its is not necessary (i.e., if local versions of all SGs are newer than remote ones). α and β both check the content of the offer messages they received. α keeps ready to receive an update message from β since all the content of the offer message are similar and newer of the SGs it has in its cache. β, on the other hand, starts to create an update message concerning SG_A. Through this update message, β informs α that it should *"add"*

descriptors *A4* and *A5* to SG_A to update it from *v3* to *v4*, with both descriptors included in the update message.

Conversely, there is a risk that two (or more) gossiping phases, involving one device and distinct new neighbors, interleave with the result that two neighbors eventually send to the same device an update message concerning the same most recent SG. But it should be noticed that, on the one hand, this redundant message is not harmful and will be ignored by the recipient, and, on the other hand, this interleaving is not probable if, like in our implementation, the beaconing used for the neighboring discovery is asynchronous with a period of several seconds. On the contrary, striving to avoid this redundant messages would impose a costly synchronization of the gossiping phase.

At the reception of the update message, α stores the hash key of SG_A with the hash keys of the included descriptors in $OSG_{\mathbf{A}}$ for future comparison. Indeed, this operation is performed to reduce the amount of data transmitted in the future update messages. A message containing only the differences between two versions of a SG will have a smaller size than a message containing the whole SG. Next, α scans the different entries included in the update message it received from β. It adds the new descriptors *A4* and *A5* to SG_A (and eliminates the outdated ones if specified). Finally, α generates the new hash key of SG_A, which must match β's hash key. In general, this operation is repeated on all the SGs of different clusters included in the update message.

It is worth noticing that the gossiping phase of TAO-DIS is very light, essentially because we exploit the fact that the messages to be disseminated are versioned catalogs of descriptors originated from a limited number of senders:

- Offer messages only contains a set of hash keys (one per cluster). This must be compared to summary vectors used in epidemic protocols, which are lists —of message identifiers— whose length can be important.
- The number of exchanged messages between two encountering devices is reduced to two or three in most cases (one offer message and one SG update message in one or each direction). In a stable situation, that is, when the encountering devices have already the same knowledge regarding the available services, only one or two short offer messages are transmitted.

5.4 Discussion and Conclusion

In this chapter, we have presented TAO-DIS protocol, the protocol related to the service discovery problem in ICHNs. Since our aim is to fully disseminate the *SG*s on all potential clients in the network, TAO-DIS introduces an optimization on the fully epidemic protocol. It takes into account the fact that service providers are hosted by infostations in order to drastically reduce the amount of data exchanged for disseminating service descriptors among the mobile devices. In our proposition, we want to reduce to minimum both the number of exchanged messages and the amount of exchanged data in the network, while preserving the same dissemination delay as that obtained with the fully epidemic protocol. A detailed evaluation of the proposed solution is presented in Chapter 8.

6

TAO: a Protocol for Service Invocation in ICHNs

In this chapter, we present TAO-INV, the part of the platform that supports the invocation phase. TAO-INV is responsible of routing the service requests an responses between mobile clients and infostations after the discovery phase. The main objective here is to preserve scalability, robustness and efficiency throughout the service invocation process in ICHNs. This chapter details TAO-INV protocol and the mechanisms it relies on.

6.1 Overview of TAO-INV

Following the discovery phase supported by TAO-DIS, we present in this Chapter TAO-INV, the protocol responsible of performing service invocation in ICHN.

The main goal of proposing TAO-INV is performing a simple but efficient service invocation in the special ICHN networks. For that, in the design of TAO-INV we combine the simplicity of counter-based protocols with the efficiency and performance of utility-based protocols in disseminating messages in ICMANETs.

The originality of TAO-INV resides in the adaptation of several well known heuristics and mechanisms to the context of service invocation in ICHN, and their combination in a coherent platform. In particular, TAO-INV implements a form of anycast for service request messages that carefully mixes a utility-based forwarding epidemic (based on the undermentioned timestamping-based heuristic) that is efficient in rapidly propagating a message, with counter-based rules aiming at reducing the number of copies spread in the network. In addition, specific attention has been paid to avoid network congestion by eliminating additional phases before message forwarding (e.g., the exchange of summary vectors) and for reducing computation by keeping our heuristic very simple compared to approaches based on context and prediction (especially on social-oriented ones), or network coding.

As explained, we assume that all the infostations belonging to the same cluster are connected to each other using a backbone. Therefore, an infostation can provide services itself, or can act as a proxy to another infostation. This destination specified in a service request does not include an infostation address but consists only in a couple (service id, cluster id). So, within a cluster, any infostation is able to serve a request. This situation conveys a differentiation in the role played by an infostation in the invocation process, according to the fact that this infostation belongs to the cluster specified in the request or

not. Actually, all the infostations that do not belong to the targeted cluster will behave as ordinary mobile devices (except that they do not move!).

6.2 Protocol Specification

Limiting the number of forwarders while maintaining high performance is a major issue for most routing protocols. In ICHN, the intermittent connection constraint makes it difficult and costly to collect necessary information for taking routing decisions. For that, we need an efficient mechanism that tries to select the best next message forwarders among a set of neighbors based on the least possible information extracted from the network.

In our work, we rely on *timestamping* to classify neighbors as good or bad carriers. This classification aims at estimating the ability of these intermediate nodes to deliver a message to an infostation. Basically, a neighbor is classified as a good carrier if its contact date with an infostation is more recent than the current carrier's contact date with an infostation. Otherwise, the node is classified as a bad carrier (a formal definition will be given later). This classification process has a dynamic property as it changes depending on the event the carrier node is passing through. A list of these events will be presented in the next section.

To elaborate, the carrier node gives a copy of the message to an arbitrary fixed number of neighbors having had the most recent contact time with an infostation. For this, a node memorizes the elapsed time since its last contact with the infostations and records temporarily the same kind of information obtained from its neighbors, thanks to a background beaconing performed periodically by each node. The algorithms using timestamping only make use of relative times, and so the nodes' clocks do not need to be precisely synchronized.

Beside the timestamping classification mechanism implemented in TAO-INV, we introduce several heuristics to cope with the constraints imposed by the nature of ICHNs. The utilized heuristics are the following:

Multiple-copy message forwarding: In order to improve the service delivery and to avoid the worst carrier dilemma (i.e., delivering the message to a mobile node abruptly moving in the wrong way), TAO-INV implements a multiple-copy message forwarding algorithm. The source is first expected to forward a copy of this message to its best neighbors. Later these carriers, and the source node, will forward a copy of this message only when they encounter a *better carrier* than themselves, while a limited number of copies will be forwarded to *bad carriers*. Furthermore, sometimes it might be risky to forward copies exclusively to good carriers, especially if the source node is located remotely from an infostation. In this case, the major number of neighbors might be classified as bad carriers, while the contact with a good carrier is uncertain. Thus in TAO-INV, each mobile node has a stock of a few number of copies dedicated to bad neighbors. This number of copies is limited in order to avoid network overload and resource consumption on mobile devices. Finally, TAO-INV makes it possible to control the propagation of messages in the network using two parameters: a lifetime and a number of hops. When the lifetime is expired or when the number of hops is zero, the message is removed from the local cache and will not be forwarded anymore.

***Network healing*:** In order to reduce this dissemination of service requests in the network, we have implemented in TAO-INV a network healing mechanism that exploits the request/response paradigm. When a carrier receives a response for a request it has locally, it removes the request from its local cache and it stops to forward this request. Thus progressively, the request will no longer be forwarded in the network.

***Source routing forwarding*:** TAO-INV is able to exploit end-to-end routes when they exist, reducing the propagation time and the number of message copies. A service response will indeed follow the reverse route of the corresponding request, as long as it is possible. If the source routing fails, because an intermediate node becomes unreachable, the message carrier will ask its direct neighbors about any node that was in good position on the reverse route so it can forward the response to it.

6.3 Node State

The TAO-INV main heuristic relies on the maintenance, on node i, on the one hand of a list D_i of elapsed times since its own last contacts with the different infostations belonging to the same cluster, and on the other hand of a list E_i of elapsed times since the last contacts of its neighbors with these infostations. When node i must forward a service request toward a service provider, it is expected to compare D_i (of a specific cluster) with E_i in order to select the best next forwarder(s)/carrier(s) to deliver the message to an infostation. A neighbor j will be considered as a *good* carrier by node i if

$$\begin{cases} \forall\, k \neq j \ and \ k \in N_i,\ E_i[j] < E_i[k] \\ and \\ E_i[j] < Min(D_i) \end{cases}$$

where N_i contains the one-hop neighbors of node i (with the date of reception o their last beacon attached to each of them). Mobile nodes or infostations are considered disconnected from the neighborhood of a mobile node if no beacons have been received from them during a gap superior to a predefined disconnection time threshold. $E_i[j]$ is equal to Min(D_j), which is the minimum value of the elapsed times since the last contacts of node j with the different infostations. Min(D_j) is piggybacked by node j in its beacon messages, so that its neighbors can perform their selections. This value is computed for each set of infostations belonging to the same cluster, as it is sufficient to contact any infostation to invoke a service provided by one infostation in the same cluster.

Unlike PRoPHET [77], PropicMan [92] or HiBOp [9], TAO-INV needs neither to record any large set of history values nor to use complex algorithms to select the next message carriers. In TAO-INV, each node i maintains three lists: N_i , E_i , D_i. These lists contain a limited number of entries (on the order of the number of neighbors for N_i and E_i, and on the order of the number of clusters for D_i).

6.4 Routing Procedure

In this part, we focus on the main routing procedure performed by each node in the network, i.e., clients, relays and infostations. TAO-INV is a reactive and an event-driven protocol. Consequently, each node will perform a specific task as a response of one of the following five events considered in the invocation process of TAO-INV:

1. The arrival of a new neighbor ;
2. The disappearance of a neighbor ;
3. The emission of a service request ;
4. The reception of a service request sent by a neighbor ;
5. The reception of a service response sent by a neighbor.

6.4.1 Management of Neighborhood Changes

The first two above-mentioned events are triggered when changes occur in the one-hop neighborhood of a node. When a new neighbor j joins the one-hop neighborhood of another node i, the D_i and E_i lists are updated using the pieces of information obtained from the beaconing process. Then, the current carrier i is expected to check if it exists in its local cache some messages (requests or responses) that should be delivered. If so, it conditionally forwards copies of these message to its new neighbor using Algorithm 6.1. If the new neighbor j has an elapsed time since its last contact with any infostation (belonging to the same cluster of the infostation providing the service) smaller than those of the other neighbors, or smaller than those of carrier i itself, carrier i will forward to this new neighbor all the service requests it has and that are still valid. The service responses stored locally will be forwarded by the current carrier to this new neighbor only if this one is the destination or has been used as an intermediate node to forward the request (i.e., if this node appears in the reverse source path in the header of the response). The decision of forwarding copies of the service requests or responses is described in details in the following section.

Let us consider a simple scenario to explain in details how such a process occurs (Figure 6.1). Considering an ICHN formed of one infostation and a group of mobile nodes roaming the network. We assume that all the service discovery process has been performed by TAO-DIS and all the nodes have the most recent version of the SG. Thus, any node can invoke one (or more) of the provided services by the infostation.

Let node *a* be a mobile node present in the network with node *d* entering its communication range. Receiving a beacon messages from node *d* is the first event node *a* will react to. Node *a* will detect that *d* is a new neighbor since its ID is not recorded in the N_a list. As a consequence, node *a* will record the ID of node *d* in this list and the date

Algorithm 6.1 Management of service messages after contacting a new neighbor.

Forwarding service messages to the newly arriving neighbor:

```
1: if event is arrival then
2:   if the new neighbor is a good carrier or size of stock > 0 then
3:     for all valid request in the local cache do
4:       Forward the request to the new neighbor
5:     end for
6:     if the neighbor is a bad carrier and the size of stock > 0 then
7:       Decrement the stock
8:     end if
9:   end if
10:  for all valid response in the local cache do
11:    run algorithm 4
12:  end for
13: end if
```

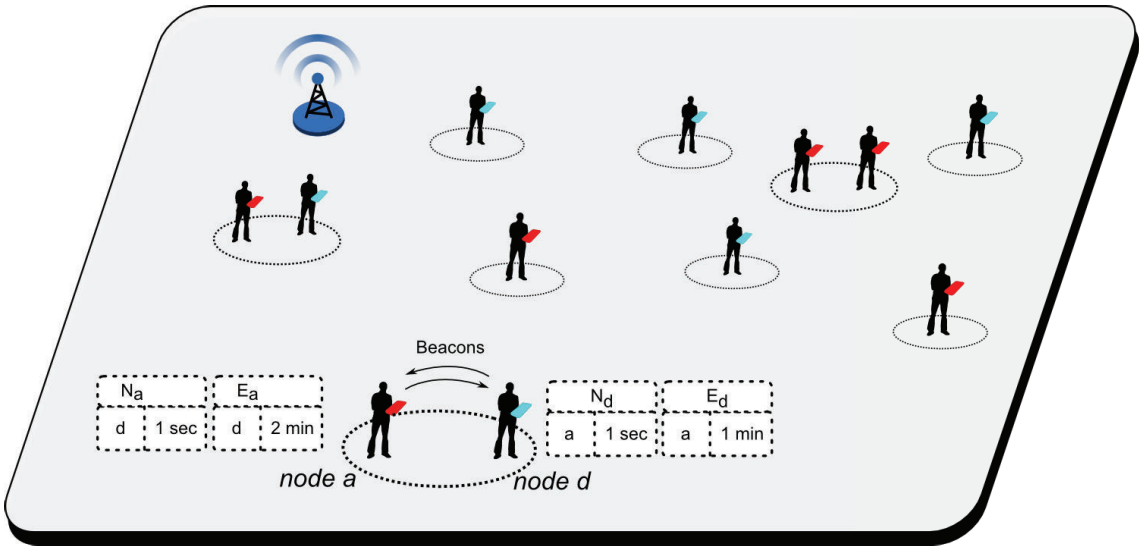


Figure 6.1: Simple scenario of service invocation in ICHN

of reception of the beacon message to compute the elapsed time since the last received beacon. In addition, node a will extract the value of the elapsed time since the contact of node d with the infostation (i.e., Min(D_d)) and introduce it in E_a list. Simultaneously, node d will also perform the same process locally. This process is followed by forwarding the valid service requests and responses that node d is entitled to obtain copies of.

As Algorithm 6.2 shows, when a node i is notified of the disappearance of an infostation from its one-hop neighborhood (i.e., from the neighbor set N_i), it updates its D_i list with the ID and the value of the elapsed time since the reception of the last beacon from this infostation. Otherwise, if the disconnected neighbor is a mobile node, all the related information to this node will be removed from the N_i and E_i lists.

Similarly to the previous example, let node b be a mobile device present in the communication range of the infostation. When node b moves out of the vicinity of the infostation, the disappearance of the infostation is detected by the interruption of the reception

Algorithm 6.2 Management of the disappearance of a neighbor.

Disappearance of a neighbor:
1: **if** event is disappearance **then**
2: **if** the node is an infostation **then**
3: Update the $\mathcal{D}_i$ list with the elapses time since the last beacon and the ID values of the infostation included in the $\mathcal{N}_i$ set
4: **end if**
5: **if** the node is a mobile node **then**
6: Update the E_i by eliminating the ID and elapsed time values of the respective disconnecting neighbor
7: **end if**
8: **end if**

of the beacons from the infostation. As a consequence, node b will record the date of reception of the last beacon to compute, when needed, the elapsed time since the last contact with the infostation. This information is registered in the D_b list to be used later by node b or its neighbors.

6.4.2 Forwarding of Service Invocation Requests

When a node i receives a service request from a local application for a service provided by a remote infostation (event 3), or when it receives such a request from one of its neighbors (event 4), node i will process this request according to the forwarding Algorithm 6.3. It will forward a copy of this request to E_{emit} direct neighbors at most in a specific instant, favoring the good carriers to the detriment of the bad carriers. When node i forwards a copy to a good carrier, it registers this piece of information to this service request. Thus, when a new node j appears in its neighborhood, node i will be able to compare Min(D_j) (the minimum value of the elapsed times since the last contacts of node j with the infostations belonging to the same cluster of the infostation providing the service) with the pieces of information registered in the service request, and thus to decide if the new neighbor j is a better carrier than itself or than the previous good carriers.

On the other hand, bad carriers will receive a copy from the stock of copies dedicated to them only when the number of good carriers is less than E_{emit} in the neighborhood of the carrier node. The stock will be decremented, and when this stock is empty no more copies of the message will be forwarded to the bad carriers.

When a node discovers that it exists in its one-hop neighborhood an infostation, it forwards to this infostation all the service requests it has in its cache specific to the cluster the infostation belongs to and are still valid, and removes them from its cache of messages. This infostation is expected to either deliver the service itself or forward the request to the appropriate infostation in the wired part of the ICHN. Any service request copy will stay alive until its lifetime expires or its number of hops is zero. Moreover, each message will store in its header the route it followed to reach the infostation.

Algorithm 6.3 Emission or forwarding of an service request.

Emission or forwarding of a Service Request:
Data:
$\mathcal{N}_i$: neighbor set of node i
Cluster $\mathcal{A}$: cluster of the infostation providing the service
if it exists an infostation in $\mathcal{N}_i$ / infostation $\in$ Cluster $\mathcal{A}$ **then**
  Forward Service Request to the infostation
  Remove the Service Request from the cache
else
  check E_i list for good carriers
  while number of emmisions is less than Min(E_{emit}, sizeof($\mathcal{N}_i$) **do**
    if good carriers are found **then**
      Forward a copy of the Service Request
      Update the best elapsed time with an infostation relative to this Service Request
    else
      if local stock related to this Service Request > 0 **then**
        Forward a copy of the Service Request
        Decrement the stock
      end if
    end if
  end while
end if

Applying this on our example, considering that node *a* wants to invoke a service at time t_2. Providing sample values of the different parameters of TAO-INV we propose that:

- E_{emit} = 3 , thus a node can at most send three copies of the service request to three neighbors in a specific instant.
- *Stock* = 3, three copies of the service request can be forwarded to bad carriers.

As shown in the scenario, at t_2, node *a* has four neighbors *b*, *c*, *d* and *e*. Thus, only three of these neighbors can obtain copies of the service request. When a node joins the neighborhood of any other node, it includes in its beacon the elapsed time since the last contact with the infostation (i.e., Min(D_i)), all these values are registered in the list E_i found in the local cache. In our case, D_a = 5 min, and the values registered in E_a are 1 min, 2 min, 10 min and 15 min for nodes *b*, *c*, *d* and *e* respectively. Consequently, nodes *b* and *c* are considered as good neighbors (both elapsed times since their last contacts with the infostation are less than the elapsed time since node *a* last contacted the infostation) and will get copies of the service request.

Additionally, since E_{emit} = 3 and we only have two good carriers and since the stock of this exact request is > 0, thus node *d* (which is classified as a bad carrier) will also get a copy of the service request. Although, nodes *d* and *e* are both bad carriers but since node *d* has a less elapsed time, thus it is considered as the best of the rest and is chosen to be the carrier of this service request. Finally, the addresses of *b*, *c*, and *d* will be registered in the header of the three service requests before being forwarded.

It is important to note that after meeting a good carrier, the threshold of classification of the new neighbors between good and bad carriers will not be performed according to

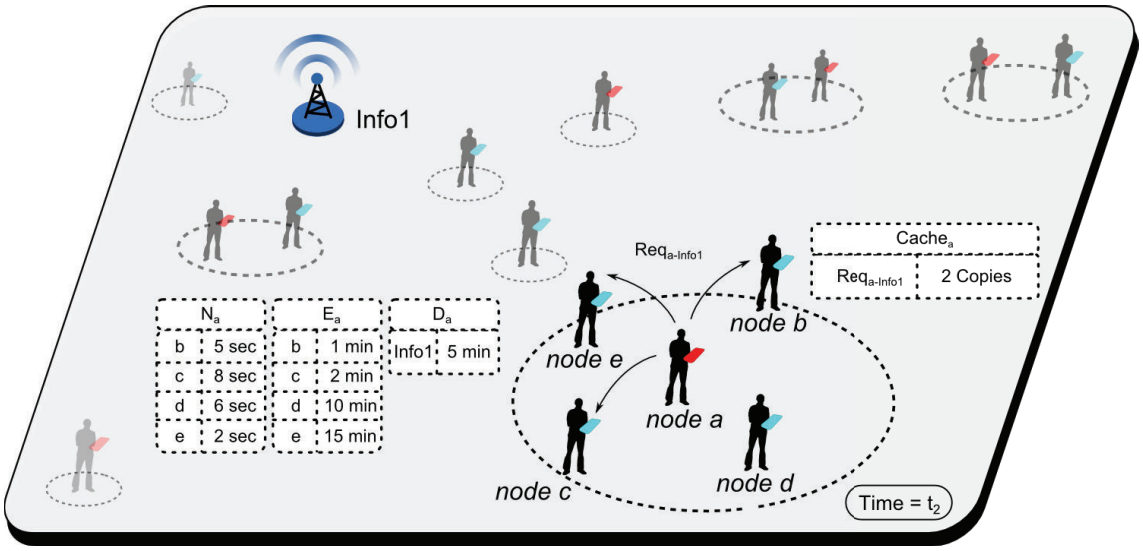


Figure 6.2: Service request forwarding in ICHN

the value of D_a (which is = 5 min), but according to the value of D_i of the good carrier, in our case D_b (which is = 1 min). Thus, for a new node to be classified as a good carrier (for this exact service request), it must have an elapse time < 1 min. By this, TAO-INV eliminates the dilemma of forwarding multiple copies of the same service request to the same node frequently joining and leaving the neighborhood and in the same time limiting the dissemination of the service requests to those recently contacted the infostation. This threshold value is registered in the service request message and transferred toward the new carriers to perform their evaluations according to it and will be updated with the value of best carrier.

6.4.3 Forwarding of Service Response

The next event is the reception of the response from the infostation toward the client node. The infostation will send the service response with a reverse routing, in other words, send the message back on the route it has just traversed. Reverse source routing can be reliable only if the mobility of the nodes in the network is relatively slow. Since TAO-INV is designed to function in an ICHN, it takes into consideration the volatility of mobile nodes and the lack of end-to-end paths between nodes in such networks. For that, when a disconnection if the source routing is detected, TAO-INV applies a forwarding mechanism to overcome such disconnections.

When such a disconnection occurs, the node carrying the message will try to deliver a copy of this message to one of the nodes recorded in the header of the message, favoring the destination or the nodes closer to this one (Algorithm 6.4). So, at the reception of the service response, when the node detects that non of the nodes registered in the header of the message is found in its one hop neighborhood, it attempts to overcome this disconnection by sending a message, a "flare packet", to try to search for one of these desired nodes. The node will send this message to its two-hop neighbors asking for the presence of one of the desired nodes in their neighborhood. This process is performed only once

Algorithm 6.4 Management of the forwarding mechanism of the service response.

```
1: if the original-service-requester is connected then
2:   Forward the Service Response to the original-service-requester
3:   Remove the Service Response from cache
4: else
5:   if next ID recorded in header is connected (reverse source routing) then
6:     Forward a copy of the Service Response
7:   else
8:     Scan N_i list for a node of ID recorded in the header
9:     if ID found in the neighborhood then
10:      Forward a copy of the Service Response
11:      Remove the ID of the node from the header of the message registered in the cache
12:    else
13:      send flare message including recorded IDs
14:      if response to flare message received (ID found) then
15:        Forward a copy of the Service Response through intermediate relays
16:        Remove the ID of the node from the header of the message registered in the cache
17:      end if
18:    end if
19:  end if
20: end if
```

when the disconnection is detected. If one of the nodes is found, a copy is forwarded to it through the direct neighbor that replied to the flare packet, to resume the source routing process. Otherwise, the node will forward a copy of the message when they encounter a closer carrier to the destination than themselves (Again, the closeness of the carrier is determined according to the route information recorded in the message header).

In Figure 6.3, we introduce a simple example of a disconnection in the source routing. Lets assume that node a generated a service invocation request that traversed nodes b, c, d and e to reach the infostation. The service response generated by the infostation reached node e. By this time, node d had disappeared from the neighborhood of node e. Thus, node e creates a flare message and sends it to its two-hop neighbors in the aim of finding any of the yet to be traversed nodes (i.e., the nodes that participated in forwarding the invocation message toward the infostation and have not received the response of this exact message yet). In this case nodes b, c and d. Since node d is found in the two-hop neighborhood of node e, a copy of the response message is forwarded toward it, which in turn forwards the response toward node a.

When an intermediate node receives a response from a neighbor node, it removes the service request associated with the response from its local cache of messages, if this request exists. Doing so, this intermediate node will not forward this request any longer when it encounters new nodes. It must be noticed that, upon the reception of a service request, each intermediate node will check, before storing the request locally, if it has in its local cache a response associated with this request. If so, it will not store this request locally and will not forward it.

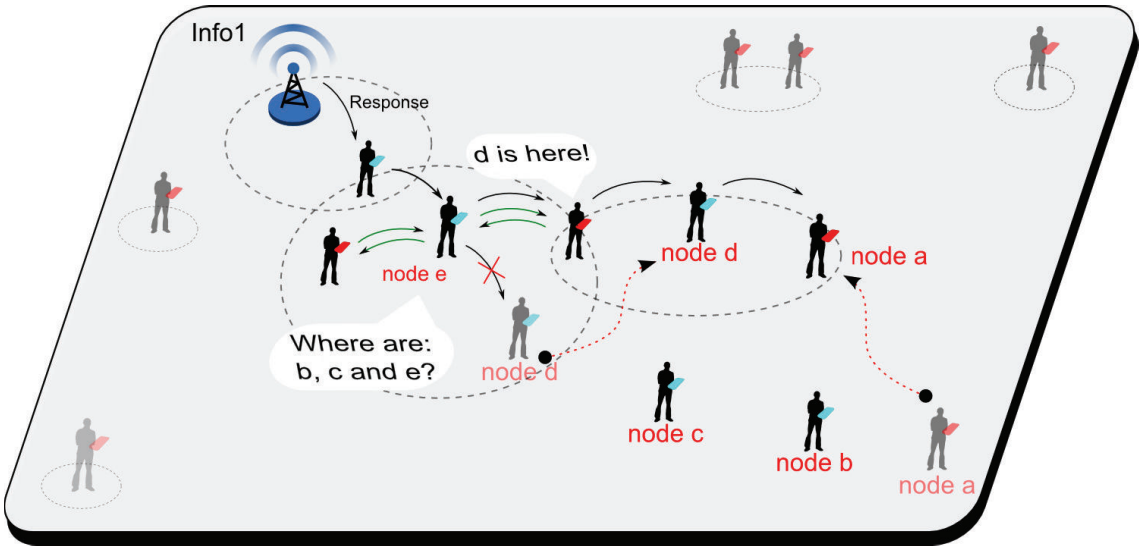


Figure 6.3: Service response forwarding in an ICHN

6.5 Discussion and Conclusion

In this chapter, we have presented TAO-INV, the protocol related to the service invocation problem in ICHNs. For that, we have succeeded in mixing a counter-based protocol with a utility-based protocol and designing a routing protocol specifically designed to target an ICHN. TAO-INV aims to benefit from the special structure of the ICHN networks to perform an efficient service invocation depending on the classification of the one-hop neighbors as good or bad carriers. By that, the protocol is able to overcome the disconnections in the network and deliver service requests from mobile clients to infostations. As for service responses, these messages are routed back to the mobile clients depending on the information collected by their relative requests. A detailed evaluation of the proposed solution is presented in Chapter 8.

7

Access Continuity in Intermittently Connected Hybrid Networks: A Handover Solution

In this chapter, we introduce an optimization over both infostation and path selection mechanisms performed in the infrastructure part of an ICHN. The main objective is to introduce a handover mechanism between the connected infostations, in the aim of improving service delivery for nomadic people and thus access continuity for clients over the provided services, while taking into consideration the opportunistic behavior of mobile clients in the network.

7.1 A Soft Handover Overview

We believe that, the presence of clusters of infostations in the network leaves room for introducing several improvements on the service delivery process. Having a backbone of infostations distributed in the physical environment can noticeably improve service accessibility on one hand, and TAO's efficiency by reducing delay and overhead values on the other hand.

Handover mechanisms designed for cellular networks ignore the opportunistic nature of communications among the mobile nodes in an ICHN. Indeed, the handover decisions are taking according to the quality of the radio signal between a base station and a mobile client. For that, an intelligent handover mechanism that takes into consideration the unconstrained mobility patterns of mobile clients and the quality of the multi-hop discontinuous paths between a client and infostation should be introduced.

For example, as we can see in Figure 7.1, at time t_0 *Bob* sent a service request message to *Alice* to forward it toward the infostation (*Info1*). By the time *Alice* moved to the vicinity of *Info1* and passed to request to the infostation (time $t_1 > t_0$), *Bob* was moving and became in the vicinity of *Info2*, which is an infostation in the same cluster as *Info1*. If no handover mechanism was applied, the response might reach *Bob* if he met *Alice* again, but certainly would need a longer duration. By implementing a handover mechanism in the infostations, *Info1* would be able to immediately forward the response toward *Info2* that in turn would forward the response to *Bob* through other intermediate mobile nodes.

In the rest of this chapter, we present the soft horizontal handover mechanism we have designed and implemented in the infostations in order to improve the service delivery for nomadic people. Unlike the handover mechanism designed for cellular networks, the handover mechanism we propose takes the opportunistic nature of the communica-

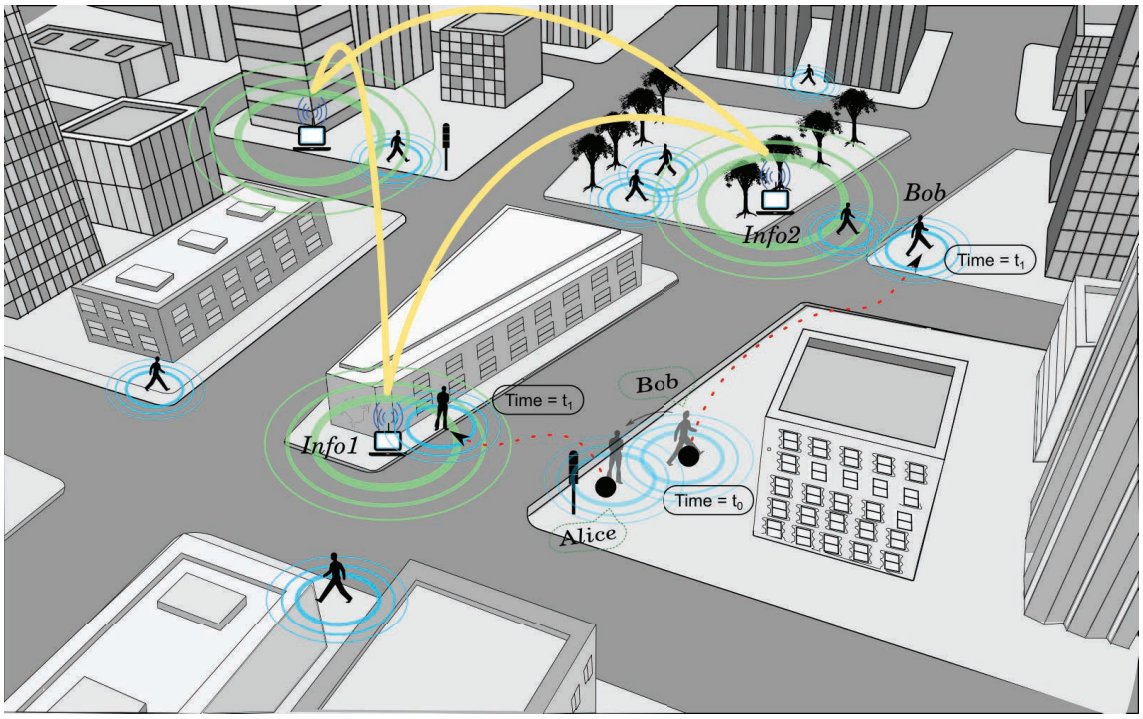


Figure 7.1: Simple scenario of handover in ICHN

tions into account. Indeed, the handover decisions are not taken according to the quality of the radio signal between a base station and a mobile client, but according to the quality of the multi-hop discontinuous paths between a client and an infostation. These paths, which can evolve dynamically according to the mobility and the volatility of the devices, are characterized by several properties, such as their stability or their length.

7.2 Infostation Infrastructures

In order to provide mobile clients with an enhanced service access, the infostations must estimate the "quality" of the discontinuous/disconnected paths (DPs) between themselves and the clients that require a service, must compare their estimations with those computed by other infostations, and, if necessary, must update their routing table according to these new estimations.

This optimization is double threaded as it can be applied on both infostations and clients sides:

- *Infostations side*: As a result of the proposed optimization, the infostations implement a soft handover mechanism to decides which infostation(s) is (are) the best to forward the service response(s) back toward a specific mobile client. Furthermore, an evaluation of the quality of the paths between the infostation and the mobile client is performed, thus the number of copies sent back is decided based on the path quality estimation.
- *Clients side*: As a result of the proposed optimization, the clients will be able to

reduce the number of copies of service requests based on the path quality between themselves and the infostations providing the services. Indeed, if the path quality is higher than a specific threshold, the number of forwarded copies for both good and bad carriers will be drastically reduced.

In the handover solution we have designed, these estimations are obtained by the infostations by processing the pieces of information stored in the service invocation requests they receive, such as the date of emission, the lifetime, the location of the client, etc. Similarly estimations are obtained by the mobile clients by processing the pieces of information stored in the service responses they receive from the infostations. The computation algorithm and the properties we consider are detailed later in this chapter**.**

The handover solution works as follows: When the infostation receives an invocation request from a new client, or when they compute an estimation that is better than the previous estimation they have in their routing table, they update their routing information and exchange summary vectors with the other infostations in the same cluster in order to allow them to update their own routing tables in turn.

The infostations are likely to not receive requests from a given client during a long period, because he has moved away, has became isolated, or has been simply switched off. Thus, the information about this client must no longer be stored in the routing table of the infostations. So as to cope with this issue and to maintain only the recent connections with mobile clients in the tables, we assign a date of computation and a lifetime to each entry. All the infostations thus share the same perception of the infostation(s) that must forward the responses to a given client.

In some situations, two (or more) infostations can approximately compute the same estimations for a given client. These infostations are therefore considered as equivalent for the service provision, and all of them should forward the service responses to the client, thus implementing a soft handover mechanism. In the remainder of this section, we describe how this handover solution operates with the both phases (request and response) of the service invocation process.

Service Invocation

Invoking a service in an ICHN basically consists in forwarding an invocation request toward a given infostation, which in turn will process the request itself if it provides the required service, or will forward the request to the infostation that provides the service.

Service request messages are forwarded using TAO-INV's timestamping-based heuristic and service response messages are forwarded using source routing techniques. While being forwarded, the messages are updated in order to include the IDs of the intermediate nodes, as well as the other properties that will allow to estimate the quality of the discontinuous path. This list of IDs will then be used in order to compute the reverse route.

In the solution we propose, the clients can process the pieces of information stored in the service responses they receive with an algorithm similar to that implemented in the handover mechanism so as to evaluate the quality of the DPs and to select the best DP, and thus additionally limiting the overhead resulting from the service request phase.

The estimation of the quality of DP between the client and the infostation is based on the value of the recorded timestamp (TAO's heuristic). Then, it chooses the best reverse DP(s) that must be followed to forward an invocation. Sometimes several DPs are considered reliable enough, only one of these candidate paths is selected (the best one). Otherwise, the message will be forwarded following TAO's algorithm introduced in the previous chapter.

When forwarding their responses toward the clients, the infostation use the reverse route defined in the invocation message. When the source routing fails because an intermediate node is no longer reachable, the intermediate node that has detected the failure will execute the same algorithm as the initial client, thus dynamically updating the DP.

7.3 Handover Mechanism for Opportunistic Computing

Handover decisions and route selections rely on the estimations of the "quality" of the DPs. In the solution we propose, the DPs are characterized in terms of message propagation time, of distance and of stability.

7.3.1 Message Propagation Time

The propagation time is an important metric in the service provisioning process. It reflects the quality of service that is directly perceived by the end-users in terms of reactiveness. The propagation time is simply the time needed by the message to propagate through the network from the message source toward the message destination. The propagation time is computed either by the recipient or the destination of the message (i.e., by a mobile client or an infostation). The propagation time for a message m is given by:

$$pt(m,t) = t - m[de] \tag{7.1}$$

Where t is the date of reception of message m and $m[de]$ is the date of emission of message m.

7.3.2 Distance

The second metric considered for the handover decisions is the distance separating a source node from an infostation. We consider two different expressions of the notion of distance: a geographical distance and an estimation of the physical distance based on the number of hops between a source node and an infostation.

We use the *haversine* formula[1] to calculate the geographical distance between a client and an infostation. The haversine formula is an equation important in navigation, giving great-circle distances between two points on a sphere from their longitudes and latitudes, i.e., the shortest distance over the earth's surface.

[1] http://mathforum.org/library/drmath/view/51879.html

Thus, the geographical distance between a client and an infostation is given by:

$$d'(m) = R \times \arccos(\sin(m[lat]) \times \sin(lat_I) + \cos(m[lat]) \times \cos(lat_I) \times \cos(m[lon] - lon_I)) \tag{7.2}$$

Where, $R = 6378.137\,km$, which represents the radius of the Earth, and the latitude and the longitude of the infostation and the client are respectively defined in radians by (lat_I, lon_I) and $(m[lat], m[lon])$.

For obvious reasons of energy consumption, nomadic people activate the GPS receiver of their handheld devices only episodically. In order to cope with this issue, we use another estimation of the distance based on the information collected from the network (i.e., TAO's time heuristic and messages exchanged among nodes in the network). It must be noticed that, since the clients are mobiles and the links are intermittent, a minimal number of hops between a client and an infostation does not guarantee a minimal geographical distance between these two entities.

The estimation we propose relies on the elapsed duration since the last contact between the client and the infostation (extracted from TAO-INV's heuristic). The approximation is define as follows:

$$d''(m) = s \times tg \tag{7.3}$$

Where s is the maximum movement speed of the node (typically 2 m/s for a pedestrian) and tg is the time gap since the last contact date with an infostation.

Due to the nonuniform movement of nodes in the network, this estimation looses its accuracy with time. For that, we benefit from the exchanged messages between the mobile nodes when possible to apply corrections on the estimated distance between the mobile nodes and the infostation. The estimation we propose therefore combines this number of hops between a client and an infostation with the message propagation time in order to approximate the maximum distance between these two devices. This approximation is define as follows:

$$d'''(m) = m[nh] \times CR + s \times (pt(m,t) - m[nh] \times \Delta_{PT}) \tag{7.4}$$

Where $m[nh]$ is the number of hops traveled by message m, CR is the Wi-Fi communication range (typically 80 meters), Δ_{PT} is the delay of an immediate forwarding and s is the maximum movement speed of the node (typically 2 m/s for a pedestrian). To apply this correction on the estimated distance, the creation date of the message by the infostation should be more recent than the contact time of the node with the same infostation.

The distance $d(m)$ between a mobile client and an infostation is thus given by:

$$d(m) = \begin{cases} d'(m)\ , & if\ location\ properties\ are\ available \\ d''(m)\ , & if\ contact\ with\ infostation\ is\ recent \\ d'''(m)\ , & otherwise \end{cases} \tag{7.5}$$

7.3.3 Path Stability

The stability of a DP is another important metric because it reflects the ability to efficiently forward a message to an infostation or to a mobile client using the source routing technique and the ability to recover an alternative path if the source routing fails. Consequently, we consider the number of neighbors of the intermediate nodes as an element of stability since it allows to take alternative paths if the source routing fails.

Furthermore, this stability depends on several factors, such as the mobility of the intermediate nodes, their power budget, etc. Indeed, the devices are carried and used by humans, and therefore can move freely or following social mobility patterns and can be switched on/off for energy consumption purposes.

In the current implementation of our solution, we weight each estimation with the distance of a neighbor from the considered intermediate node if the locations, the speeds and the directions are known. Otherwise, we weight these estimation with the contact times that are simply defined by:

$$c_i = np_i/np \tag{7.6}$$

Where np_i is the number of hello packets received from node i (i.e., the number of messages of presence sent by i) and np is the number of hello packets the node i is expected to send since its appearance in the vicinity of the current node. When the value of this property is equal (or close) to 1, a node i is considered as a stable neighbor of the current node. On the contrary, a value close to 0 reflects the sporadic appearance of node i in the neighborhood of the current node.

A lifetime is associated with this value so as to consider only the last contacts between two nodes. The path stability estimation obtained locally (i.e., for a given intermediate node) is thus:

$$\sum_{k=0}^{n} ns_k \ \ where \ \ ns_k = \begin{cases} d_k \ , \ \ if \ location \ properties \ are \ available \\ c_k \ , \ \ otherwise \end{cases} \tag{7.7}$$

$$and \ d_k = \begin{cases} 1 \ , \ \ if \ distanceAt(location_k, s_k, b_k, 2 \times \Delta_t) \leq CR \\ 0 \ , \ \ otherwise \end{cases} \tag{7.8}$$

Where $location_k$, s_k, and b_k are respectively the current location, the movement speed, and the bearing of neighbor node k. Δ_t is the delay needed to forward a message from the local node to an infostation, CR is the communication range of the local node and c_k is the contact time of node k. Function $distanceAt()$ returns the distance between the local node and another node at a given time based on the location, the speed and the direction of these two nodes.

As shown in Figure 7.2, lets assume that Alice is forwarding a service request toward infostation *Info1* through *node a*. For *node a* to be considered stable enough, it should be present within the communication range (CR) of Alice during double the time needed to

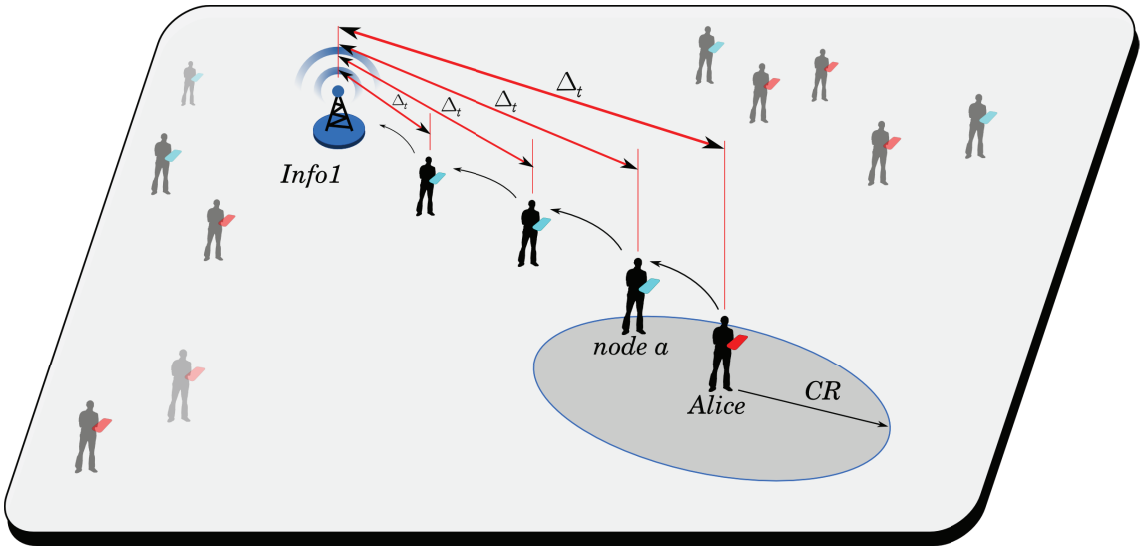


Figure 7.2: Example of path stability estimation.

forward the message from Alice to *Info1* (i.e, 2 xΔ_t) despite the mobility of both *Alice* and *node a*. By that, we ensure that when a service response is generated, the nodes that have contributed in forwarding the service request are always present in the vicinity of each other. As a result the path followed by the message is always stable and is reliable to send back the service response toward Alice. This estimation is performed locally by all the nodes acting as intermediate nodes, with all their estimations included in the service request message to be evaluated by the infostation when it receives the message.

Finally, the path stability value is the minimum of the estimations obtained along the path. It is thus defined as follows:

$$m[path\ statbility] = min(m[path\ \ stability], new\ \ estimation) \tag{7.9}$$

Where $m[path\ stability]$ is the stability of the path taken by message m. The function that returns the path stability is thus defined by:

$$s(m) = m[path\ stability] \tag{7.10}$$

7.3.4 Handover Algorithm

The handover algorithm aims at choosing the infostations that must forward the responses to a given client based on the above presented metrics.

When an infostation receives an invocation request from a client, it estimates the quality of the path taken by the invocation request. Then, it checks its routing table for the previous estimations it has for this client. If it has no information about this client, it

Algorithm 7.1 The section of the algorithm applied upon service invocation reception.

Data:
$\mathcal{R}$: the routing table — m: the incoming invocation request
I: the current infostation — D: the current date
$\mathcal{F}$: the estimation function — $\mathcal{V}$: the summary vector

$\mathcal{R} \leftarrow \mathcal{R}$ - {$\mathcal{R}$\{client=m[source] & infostation = I\}}; T $\leftarrow$ $\mathcal{R}$\{client=m[source]\}
$\mathcal{E} \leftarrow \mathcal{F}(m)$
if (T = $\emptyset$) **then**
  $\mathcal{R} \leftarrow \mathcal{R} \cup$ {m[source],I,D,$\mathcal{E}$} ; $\mathcal{V} \leftarrow$ {add,{m[source],I,D,$\mathcal{E}$}} ; send $\mathcal{V}$
else
  if ($\mathcal{E} \geq$ max(T[estimation]) **then**
    if ($\mathcal{E} \geq \Gamma_{\mathcal{E}}$) **then**
      $\mathcal{R} \leftarrow \mathcal{R} \cup$ {m[source],I,D,$\mathcal{E}$} - T
      for all k $\in$ T **do**
        $\mathcal{V} \leftarrow \mathcal{V} \cup$ {remove,k}
      end for
      $\mathcal{V} \leftarrow \mathcal{V} \cup$ {add,{m[source],I,D,$\mathcal{E}$}} ; send $\mathcal{V}$
    else
      for all k $\in$ T **do**
        if (k[estimation] + $\Delta_{\mathcal{E}} < \mathcal{E}$) **then**
          $\mathcal{R} \leftarrow \mathcal{R}$ - {k} ; $\mathcal{V} \leftarrow \mathcal{V} \cup$ {remove,k}
        end if
      end for
      $\mathcal{V} \leftarrow \mathcal{V} \cup$ {add,{m[source],I,D,$\mathcal{E}$}} ; send $\mathcal{V}$
    end if
  else
    if ($\mathcal{E}$ > max(T[estimation]) - $\Delta_{\mathcal{E}}$) **then**
      $\mathcal{R} \leftarrow \mathcal{R} \cup$ {m[source],I,D,$\mathcal{E}$} ; $\mathcal{V} \leftarrow \mathcal{V} \cup$ {add,{m[source],I,D,$\mathcal{E}$}} ; send $\mathcal{V}$
    end if
  end if
end if

stores this estimation in its own routing table and sends to the other infostations, on a multicast address, a summary vector including the modifications it operates on its routing table so that they in turn can propagate these modifications on their own routing tables (see Algorithm 7.1).

$$\mathcal{F}(m) = \alpha \times \frac{1}{pt(m,t)} \times s(m) \times \frac{1}{m[nh]} \times \frac{1}{d(m)} \tag{7.11}$$

The estimation of the "quality" of the discontinuous paths $\mathcal{F}(m)$ is computed using the function defined above. Where, $pt(m,t)$ is the message propagation time, $s(m)$ is the stability of the path taken by message m, $m[nh]$ is the number of hops traveled by message m, and $d(m)$ is the distance between the mobile client and the infostation. This function aims at privileging the paths that offer a good propagation time and stability, as well as the infostations closer to the client. α is a parameter of the function that allows to obtain results greater than 1 (typically α can be equal to 1000).

As described in Algorithm 7.1, if an infostation finds some estimations for the considered client, it checks if the new estimation is better than the previous ones. If so, it checks again if this estimation is greater than $\Gamma_{\mathcal{E}}$. If so, it removes the older estimations

and keeps only the new one. $\Gamma_{\mathcal{E}}$ is a parameter of the algorithm, such that, when an estimation is greater than $\Gamma_{\mathcal{E}}$, the path is considered as reliable and consequently it is not relevant to forward a message from two distinct infostations.

If the new estimation is less than $\Gamma_{\mathcal{E}}$ and better than the previous ones, the infostation keeps only the better estimations that are considered as equivalent (i.e., the estimations whose gap with the better estimation is less than $\Delta_{\mathcal{E}}$). A summary vector is sent to the other infostations in order to propagate the modifications.

Algorithm 7.2 The section of the algorithm applied upon summary vector reception.

Data:

$\mathcal{R}$: the routing table	m: the incoming invocation request
I: the current infostation	D: the current date
$\mathcal{F}$: the estimation function	$\mathcal{V}$: the summary vector

```
1: for all k ∈ V do
2:   if k[action] = remove then
3:     R ← R - {k}
4:   end if
5:   if k[action] = add then
6:     R ← R ∪ {k}
7:   end if
8: end for
```

When they receive a summary vector, the infostations execute the simple Algorithm 7.2, which consists of adding, removing and updating lines in the routing table.

7.4 Conclusion

In this chapter, we have presented the optimization we proposed over TAO to ensure an efficient access continuity. This optimization is performed by implementing a soft handover mechanism between the connected infostations, that prefers one infostation over another in terms of forwarding service responses toward the mobile clients. Unlike handover mechanisms designed for cellular and mesh networks, the handover we propose takes into consideration the opportunistic behavior of mobile clients in the network and takes decisions based on these movements. This mechanism relies on estimating the quality of the path between the infostations of the same cluster and the client requesting a specific service. A detailed evaluation of this handover is presented in Chapter 8.

8 Implementation and Evaluation

In this chapter, we introduce the representation of the middleware platform for service provisioning in ICHN followed by the description of the service management application programming interface (API). We also present the performance evaluation of the invocation and discovery mechanisms (i.e., TAO-INV and TAO-DIS) and the performance evaluation of the optimization we proposed over TAO-INV (i.e., the soft handover mechanism). By relying on these simulations, we aim to prove the validity of our propositions and simultaneously evaluate the protocols while tuning their different parameters. In the following, we describe the general simulation settings that are shared among the evaluation of the three proposed mechanisms, focusing on the metrics used to evaluate the performance of each protocol, then we analyze the obtained results.

8.1 Proposal for the TAO Platform

8.1.1 General Architecture

In this part of the chapter, we present the middleware platform we have designed in order to evaluate the proposition specified in the previous chapters. The architecture of this platform is depicted in Figure 8.1.

The TAO middleware is formed of two main modules: TAO-DIS module and TAO-INV module.

- TAO-DIS: This module implements the TAO-DIS protocol and provides an API to it. It contains several submodules that are responsible of performing the service discovery process. It links the different submodules with each other and with the other external modules such as the *Message dispatcher* and the *Neighboring Module*.

  - *Service Tracker*: This module is used by the various local applications when the requested service is not discovered yet (i.e., not found in the service registry). The service tracker creates a local service discovery request using the information specified by the local application (i.e., using the object implementing the method *isMatchedBy()* within the interface *ServiceTracker*), and periodically scans the service registry for new services. Whenever a matching service is found, the application is notified. Consequently, the application can trigger the invocation process of the desired service.

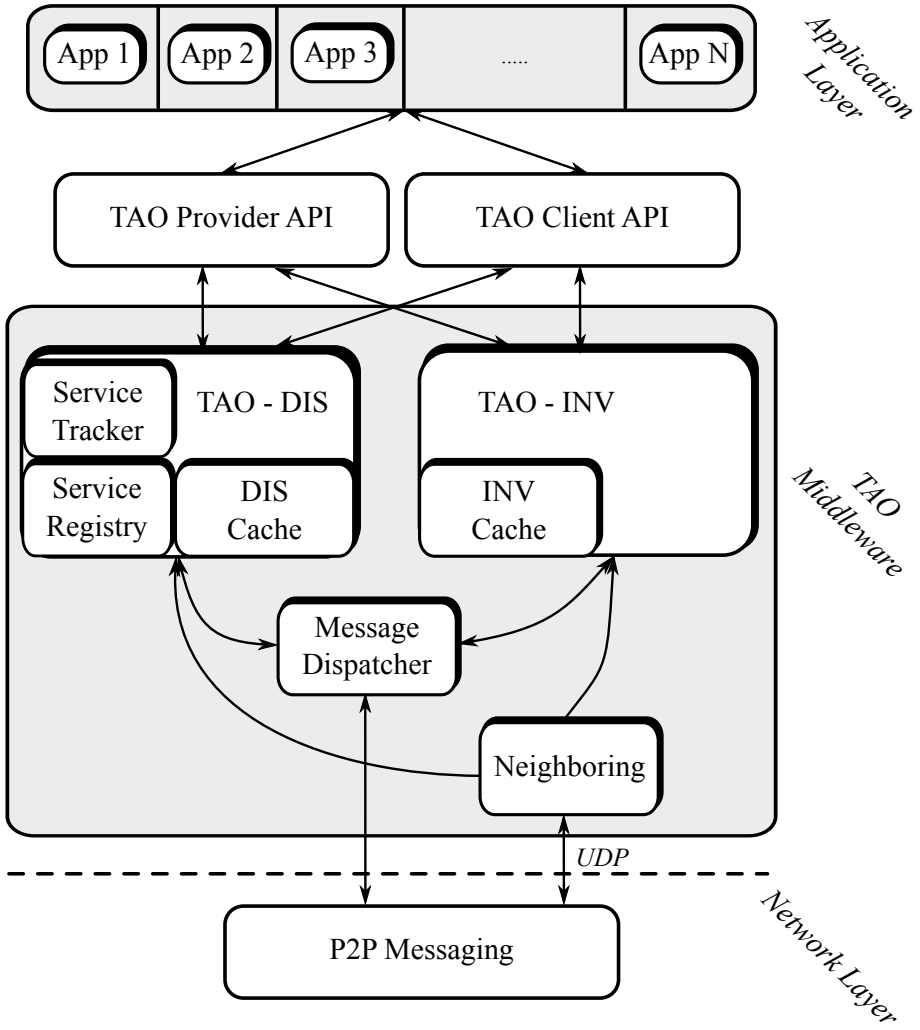


Figure 8.1: The middleware architecture

- *Service Registry*: Each mobile node in an ICHN must have its own local service registry. In such type of networks, service registries cannot be centralized but should be managed in a P2P manner. Indeed, the service registry is responsible for maintaining locally a list of information about services and service providers. The service registry is expected to be invoked by local applications in order to discover remote services and obtain references of service providers.
- *DIS Cache*: This module registers the older versions represented by the OSG list. It is accessed whenever an update should be performed on a current version of the SG or when an update should be sent toward a neighbor node.

- *TAO-INV*: Similar to TAO-DIS, this module implements the TAO-INV protocol and provides an API to it. It is responsible of the service invocation process and performs the different actions related to generating service requests, handling service responses on the client's side, handling services requests and generating service responses on the provider's side.

  - *INV Cache*: This module keeps track of the different service requests and responses generated (or forwarded) by the node.

- *Message Dispatcher*: It is the module that receives messages from the network layer. All messages such as beacons, invocation requests and responses, offer messages and SG-Updates are received by this module. Messages related to service discovery and invocation are filtered and passed toward their associated modules.

- *Neighboring*: This module informs the rest of the modules of the appearance and the disappearance of neighbors from the one-hop neighborhood. It performs these actions by managing the beacons received from the network layer. This module is also responsible of generating periodic beacons to inform the direct neighbors of the node about its presence.

As for the communication between the middleware and the network, we utilize, for the current version of the implementation, the User Datagram Protocol (UDP) [104]. In practice, the frequent disconnections due to interference and weak signals will result a large amount of retransmissions when using the Transmission Control Protocol (TCP) [120]. Besides, the sessions handling in TCP will result performance problems. For that, we utilize UDP to perform point to point communication for the current implementation of TAO.

8.1.2 APIs

The main objective behind designing this middleware is performing service provisioning in an ICHN. The provisioning process is represented by both the automatic discovery of services performed by the TAO-DIS protocol and the asynchronous invocation of services performed by the TAO-INV protocol.

The first step, before the initiation of the service discovery and invocation processes, is the creation of a service. In Table 8.1, we present the API needed by a service provider to create a service and save it in its cache. The method "*registerService*", from the *ServiceProvider* interface, takes the service name and properties (both functional and non-functional) as parameters to create the descriptor of the service. The functional and non-functional properties are used later by clients to match the pattern of the service with their needs. These information are also used to generate the hash key of the service.

Method Name	**Purpose**	**Parameters**
registerService	To create a new service	serviceName
		functional_properties
		nonfunctional_properties
recordService	To register a new service in the cache	serviceDescriptor
unregisterService	To delete an existing service	service_id

Table 8.1: API of a service provider implementing TAO-DIS

Service Discovery After the creation of the service, the service provider undergoes the publishing process of the created service. This is performed by the usage of the object implementing the interface *ServicePublisher* (detailed in Figure 8.7 in next section). The object implementing the *ServicePublisher* interface is responsible of informing other nodes roaming the network of the provided service. This object can be implemented at the service provider itself or at another infostation acting as a gateway for this service provider. As shown in Table 8.2, the publishing process is done by modifying the current SG of the infostation and forwarding the *SG-Updates* to its one-hop neighbors. This process of cache management is performed by service providers (respectively clients) at the creation of a new service (respectively informed of a new service in the network).

Method Name	**Purpose**	**Parameters**
setOfferMsg	To send the HK of local SG to one-hop neighbors	sg_hk
		version
setSGUpdate	To send missing entries of SG to direct neighbor	sgUpdateMsg
updateSG	Called at the reception of a SG update Message (client) or at the creation of a new service (provider), with action being add or delete entry	serviceDescriptor
		sg
		action
trackService	Called when an application wants to invoke a service not found in cache	service_pattern
		call_back

Table 8.2: API of cache management of TAO-DIS for both clients and providers

At the mobile nodes side, the object implementing the *ServiceGuideUpdateHandler* interface is responsible of receiving the new update messages (preceded by the exchange of offer messages) and forwarding them to the *TAO-DIS* module in order to update the current version of the SG found in the *ServiceRegistry.* This is followed by saving the previous version of the SG to the *DIS Cache*.

```
//Client searching for a specific service
...
   TaoDisClient disClient = new TaoDisCLient();
...
   Service s = disClient.lookup(service_pattern);
   String s_id = s.getServiceId();
   disClient.invokeService(s_id, method, params);
...
//case service not found, scan service registry at every SG update
   disClient.trackService(service_pattern, call_back);
}
```

Figure 8.2: Code sample - Client searching for a specific service

If a local application is expecting an undiscovered service, the object implementing the *ServiceTracker* interface will scan the new version of the SG (using the *trackService*

method) to check if the requested service can be found in the new SG. If found, the local application will be notified. Otherwise, the *ServiceTracker* will wait for a new update of the current SG. A code sample concerning this process is represented in Figure 8.2.

Service Invocation When an application is interested in one or more services found in the *ServiceRegistry*, this application can trigger the invocation process of the desired service. The API shown in Table 8.3 represents the main part needed by a client implementing TAO-INV to perform a service invocation process.

Method Name	Purpose	Parameter
getService	To choose a service among those in SG	service_pattern
asyncInvoke	To send a request to a specific infostation	requestMsg
		RespHandler
handleServiceResp	Called at the reception of a service response	responseMsg

Table 8.3: API of a client implementing TAO-INV

To invoke a specific service TAO-INV, by using the object implementing the *ServiceInvoker* interface, the client extracts the information of the desired service from the SG found in the service registry and the reference (address, clusterID) to the infostation providing the service. This is followed by generating the service request message and forwarding copies of this request to one-hop neighbors based on *good* and *bad* carrier classifications performed by the algorithms introduced in the previous chapter. A code sample that shows the invocation of a specific service by a mobile client is represented in Figure 8.3.

```
//Client invoking a service
invokeService(service_id){
   ...
   ServiceRequestMsg requestMsg = new ServiceRequestMsg();
   String method;
   Object[] params;
   requestMsg = invClient.prepareReqMsg(service_id, method, params);
   invClient.sendServiceReq(requestMsg);
}
```

Figure 8.3: Code sample - Client invoking a specific service

When an infostation receives an invocation request it directly checks if it is the original provider of this service. If not, it forwards this request within the cluster to reach the original service provider. The service provider handles this request and creates a message that includes the response. This message is forwarded toward the requesting client by depending on both the soft handover mechanism and TAO-INV. Table 8.4 includes a part of the API related to a service provider implementing TAO-INV while Figure 8.4 includes a sample code of a provider generating a service response.

Method Name	Purpose	Parameters
handleServiceReq	Called at the reception of a service request	requestMsg
getService	To search for a service in reply to a request	service_pattern
sendServiceResp	To send a response to a specific client	responseMsg
		destination_address

Table 8.4: API of a service provider implementing TAO-INV

```
//Provider responding to a service invocation
handleServiceInvocation(requestMsg){
   ...
   bool found = invProv.lookupService(service_pattern);
   if(found){
      ServiceResponseMsg responseMsg = new ServiceResponseMsg();
      Object resp;
      String resp_type;
      responseMsg = invProv.prepareRespMsg(service_id, resp, resp_type);
      invProv.sendServiceResponse(responseMsg);
   }
}
```

Figure 8.4: Code sample - Provider responding to a specific invocation request

8.1.3 Implementation Details

After presenting the service discovery and invocation processes, we describe the different interfaces of the middleware architecture. Starting with the interfaces represented in Figure 8.5:

- *OfferMessage:* It is a part of the service discovery process implemented in TAO-DIS. This message is exchanged whenever two nodes (mobile / mobile or mobile / infostation) enter the communication range of other, or when an update over the current SG of an infostation takes place (i.e., addition or elimination of a service). The exchange of this message is performed in order to keep the SGs found in the *CacheDIS* of all the nodes up-to-date.

- *ServiceGuideUpdateMessage:* It is the message sent as a consequence of the exchange of the *OfferMessage* in the case where a difference exists between the hash key values of the local and remote SGs. This message includes the entries of the SG that should be modified in order to obtain the last known version of a specific SG (i.e., relative to a specific cluster ID). Thus, such a message might contain multiple sets of entries each of which related to a specific SG.

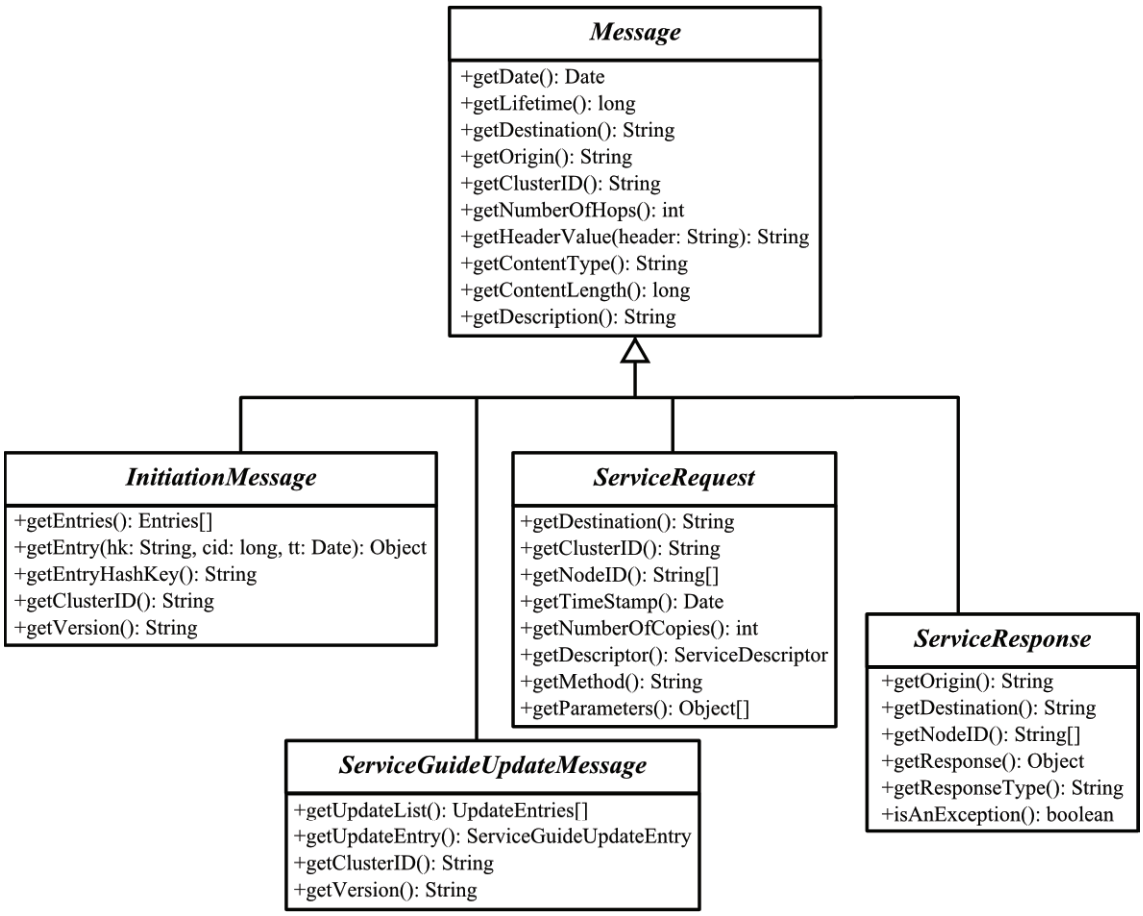


Figure 8.5: A UML representation of the message management API

- *ServiceRequest:* This message is the request sent from the client toward the infostation to invoke a specific service. It includes the service descriptor of the requested service and the method used to obtain the response from the provider according to a specific set of parameters. The rest of the methods defined in this interface are used to obtain the necessary information to perform the routing decisions of the message from the client toward the infostation (such as *NodeID, TimeStamp*, etc.).
- *ServiceResponse:* The *ServiceResponse* message is generated by the service provider as a consequence of receiving a *ServiceRequest* message. Similar to the request message, the first three methods are responsible of providing information to the algorithms to perform routing operations of the message from the infostation toward the mobile client, while the rest of the methods are related to the response, its type and if the response is generated successfully or not.

Moving to the second set of interfaces presented in the Figure 8.6:

- *ServiceRequestHandler:* This handler is implemented at the infostation side and is responsible of handling the various service request messages received from the clients

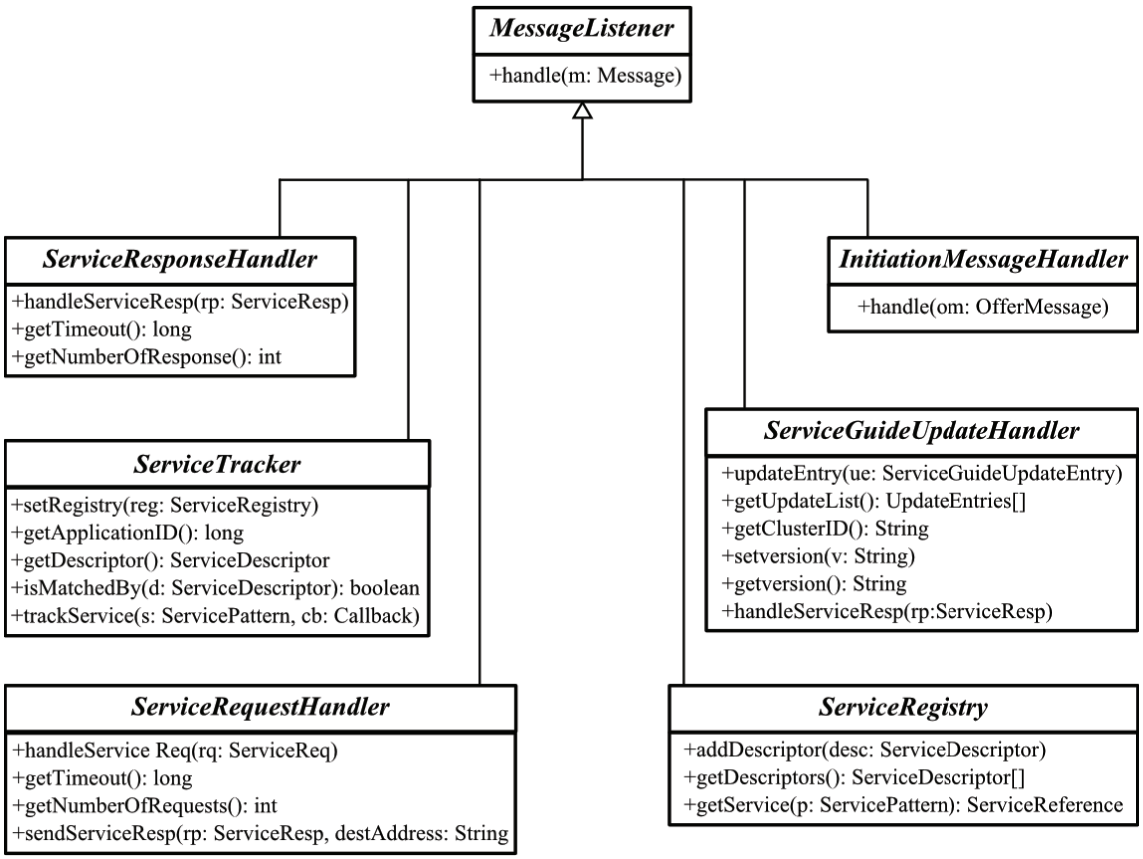


Figure 8.6: A UML representation of the service management API

roaming the network. Each handler specifies if the received request is still valid by checking the timeout value. Furthermore, it checks if the infostation should forward more than one copy of the same response to the multiple requests received (i.e., if more than one request for a specific service has been received from the same client). This is done by checking the *NumberOfResponse* value, if it is set to $n > 1$, thus n responses will be forwarded. This is done to ensure sending the same response on multiple paths to increase the probability of reaching the mobile client. n is a parameter of the protocol, specified by the service provider depending on the state of the network.

- *ServiceResponseHandler:* It is similar to the *ServiceRequestHandler* but implemented on the client side. In this handler, it is usually sufficient to set the value of *NumberOfResponse* to one, as the client is expecting one response to its request (responses for the same request received later will be discarded).

- *OfferMessageHandler:* This handler will generate a *SG-Update* message only if the local SG is more recent than the remote one; otherwise, no action will be performed.

- *ServiceTracker:* It is triggered on the reception of a new version of the SG, as it scans the *ServiceRegistry* for determining if the newly received services match the requested services by the various local applications. This is done by comparing the

service descriptors of both the newly added service and the requested ones, if a match is found, the relative application is informed.

- *ServiceGuideUpdateHandler:* At the reception of a *SG-Update* message the object implementing the *ServiceGuideUpdateHandler* interface extracts the entries included in the message to add the newly discovered services or eliminate the obsolete services from the current SG of each cluster of infostations.
- *ServiceRegistry:* The *ServiceRegistry* extracts the newly added services from each of the most recent versions of the local SGs and present them according to their descriptors (functional and non-functional properties) to become available for invocation by the local applications when needed.

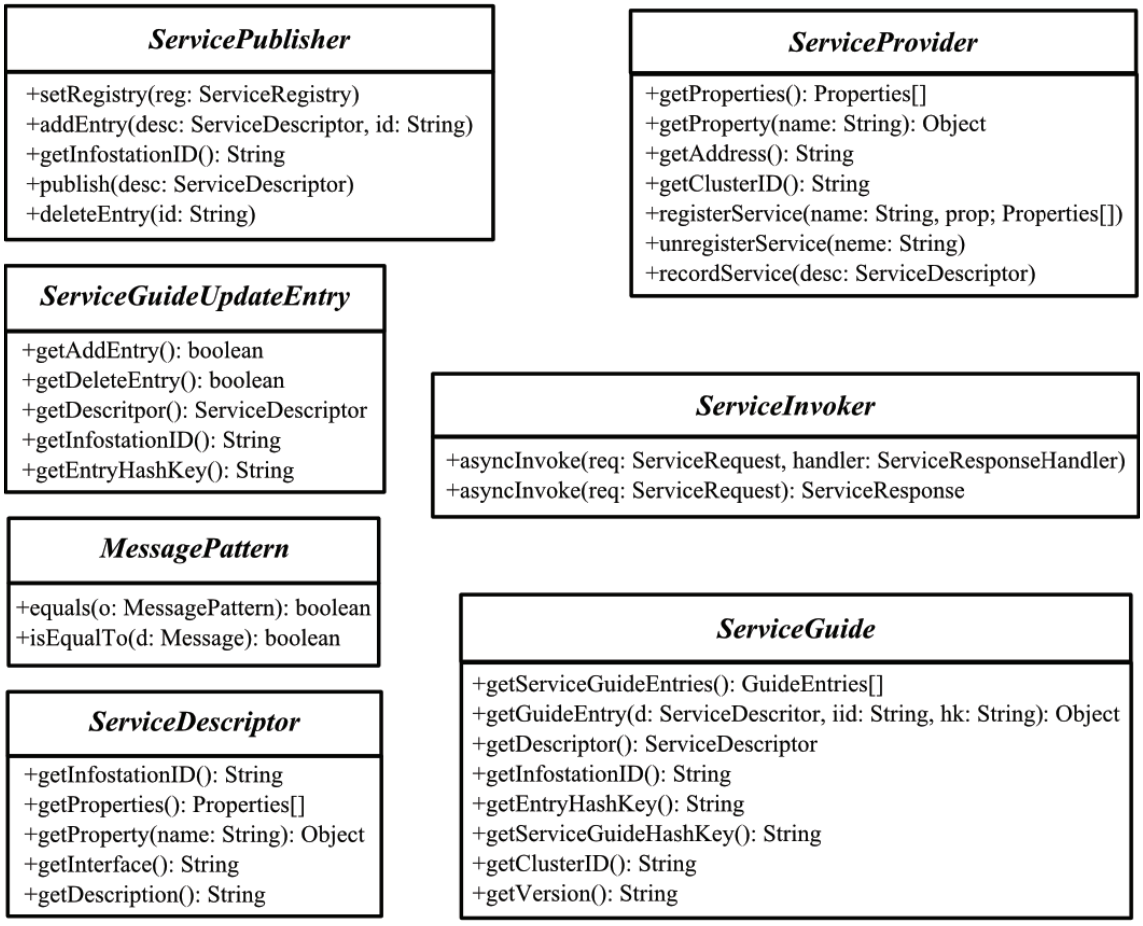


Figure 8.7: A UML representation of the invocation and discovery APIs

The final set of interfaces is depicted in Figure 8.7:

- *ServiceDescriptor:* This interface is used by the service provider to specify the functional and non-functional properties of the proposed service. It also includes the

main description of the parameters that should be included in the request message and the type of response expected.

- *MessagePattern:* The message pattern is generally used by the *MessageListener* module to detect the multiple reception of the same message. Thus, if the message received is an exact match to a previously received message, then it is automatically discarded by the *MessageListener* module.

- *ServiceProvider:* By relying on this interface, the service provider sets the values of the various properties of the service descriptor of the service and specifies if the service will be directly accessible through it or through another infostation acting as a gateway.

- *ServicePublisher:* This interface is used to modify the local SG of the infostation providing the service, and thus triggering the forwarding of *OfferMessage*s toward the one-hop neighbors to inform the mobile nodes roaming the network about the addition (or elimination) of a new service.

- *ServiceGuide:* The *ServiceGuide* interface is used to implement unique objects for each set of infostations (i.e., a cluster). This interface sums up all the information regarding the provided services by this set of infostations that are necessary to invoke any of these services. Also, it includes the current version of the SG (specified by the *version* and the *ServiceGuideHashKey* values).

- *ServiceGuideUpdateEntry:* This interface describes what each entry of the *ServiceGuideUpdate* message should look like and what should include. For example, it specifies if the update is related to adding or eliminating a specific service from the SG, beside specifying the service descriptor and the ID of the infostation providing the service.

- *ServiceInvoker:* The *ServiceInvoker* interface is used by clients to invoke a specific service found in the *ServiceRegistry* and matches the interests of a local application. With the help of this interface, the node sets a *ServiceResponseHandler* to wait for the response of the invoked service that will in turn pass this response to the requesting application. This handler is deleted after the reception of the response.

8.2 Performance Evaluation

The objective of the experiments is to evaluate the performance of the protocols we have proposed and to compare each of them with state of the art protocols under the same conditions to validate our hypotheses.

In general, the evaluation of the protocols can be performed by relying on several methods: real case studies, analytic evaluation or simulations. In general, evaluating protocols by relying on real case studies is considered very effective. However, setting a case study requires a large effort especially in controlling the different parameters of the environment. In practice, such type of evaluations is less repeatable as we cannot ensure the exact behavior of the nodes contributing in the network. In addition, with a

real experiment we are limited to a small number of devices, which affects the scalability evaluation of the protocols.

Complexity is the main constraint of analytic evaluation**.** In this type of evaluation we have to provide an analytical model of the environment. The model should include the behavior of the used applications and their effect influence on the protocol, the radio communication and the effect of interference between various devices. Adding extra constraints to approach a real case study increases the complexity of the model. In fact, we lack the tools that are able to model all the constraints present in the type of networks we are dealing with. As a consequence, if we rely on analytic analysis we would end up with a very simplistic model that does not provide a precise evaluation of the protocols.

For that, we rely on simulations to perform the evaluations of our protocols. We choose this method since it provides a wider range of freedom. By utilizing simulations we can create more complex environments than those obtained by the analytic method. For instance, in our simulations we take into consideration the mac layer and the collisions that can occur at this level. This cannot be done in an analytical model as it increases its complexity. Despite that simulations cannot perfectly represent all the events that may occur in a real case study, nevertheless, it permits us to simulate a large number of nodes and test the scalability of the protocols. Besides, it is possible to obtain a repeatable behavior of our experiments with varying specific parameters.

Various network simulators are available for the evaluation of protocols and systems of mobile ad hoc networks; among the most popular are ns-2 [47] with the so-called Monarch extension [52], Glomosim [136], the ONE simulator [55] and Opnet [46]. In general, simulators can be high level simulators, such as the ONE simulator. These simulators ignore the lower layers details and the loss of messages that might occur due to interference or collisions. Thus, neglecting such kind of details and missing various events (such as: interference, buffer overload, etc.) might result potential impact on the performance of the protocols. As a consequence, for our simulations, we have chosen OMNet++ [124]. OMNet++ is an open source modular discrete-event simulation environment and belongs to a class of tools for simulation of generic complex systems. These tools only provide primitives for the concurrent execution of multiple entities and communication among them usually by means of message passing based paradigms. By using OMNet++ we can imitate a real case study. Indeed, we can take into consideration the low-level constraints of realistic environments by including models of the MAC and physical layers. In addition, various plug-ins and extensions concerning statistical analysis of data are available for OMNet++. Finally, it is free for academic use, thus interested researches can repeat the experiments we have performed.

Simulation Environments

For the evaluation of both our propositions and the protocols used for comparison, we rely on a fixed set of parameters that characterize the different simulation environments we consider. These parameters are used to ensure the similarity in the behavior of the different nodes forming the simulated network and the targeted case study of our work. For instance, the mobility speed is kept, most of the time, within the range of walking speed. With respect to the radio technology, we assume that the communication range of mobile devices and of the infostations is set approximately to 30 meters and using

omnidirectional antennas.

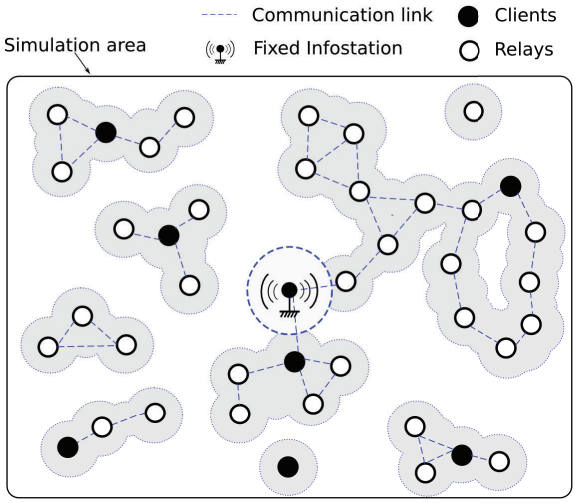


Figure 8.8: Simulation environment

Furthermore, in our experiments, we mainly focus on two sets of environments for the evaluation. The first one is based on artificial movements of mobile nodes, while the second is based on the exploitation of real movement data.

In general, when relying on artificial movements, the simulation area we consider is a square of 1 km^2 in which we define a simple hybrid network composed of one or more infostations providing a set of services and a set of mobile devices carried by pedestrians (with a speed set to 0.5m/s), as depicted in Figure 8.8. A subset of these pedestrians act as clients that request services from the infostations, the rest of the mobile devices serving as potential relays. The artificial movement of the mobile devices is determined by a *random way-point mobility model* (RWP): nodes randomly choose a speed and a destination and move to this destination at this speed. Upon arrival, the node pauses for a while and then chooses a new destination.

As for the environment based on the exploitation of real movement data, we used a dataset from the CRAWDAD repository that contains 92 GPS traces of movement of students in the KAIST campus (Daejeon, Korea) [107]. The main advantage of this dataset is that it is based on actual (GPS-based) locations of nodes and not only occurrences of contact. We built a graphical representation of the data by analyzing it and performing a projection on the actual map of the KAIST campus. The result of this projection is presented in Figure 8.9. This graphical representation of the data helped us in understanding the nature of the students' movements providing the data, and in determining the most visited places (referred to by areas with darker colors). This information was later used to choose the position of the infostations. Similar to the artificial movements environment, a subset of the students act as clients that request services from the infostations. Moreover,

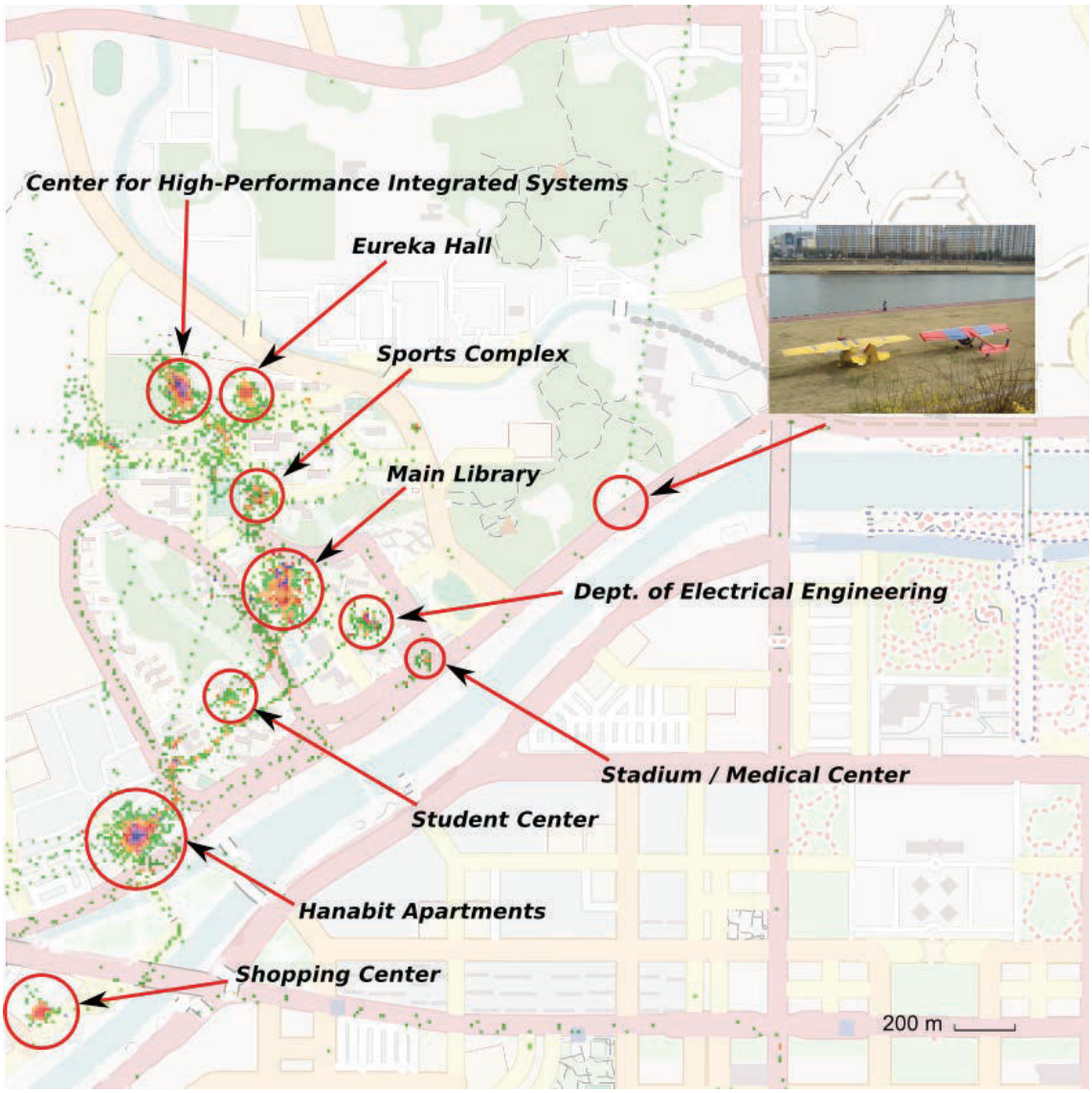


Figure 8.9: Graphical presentation of the real traces

most of the mobile nodes represent pedestrians, but some have higher a speed (students embarked in cars).

By using OMNet++, we performed different experiments to compare the performance of the protocols when utilizing the RWP mobility model or the real traces for the movements of nodes in the environment. In all our experiments we took int account the low-level constraints (the MAC and physical layers). Thus, we tried to accomplish our goal of considering most of the constraints that exist in real case studies (delays over transmission and message losses due to collisions in the various layers).

8.3 Service Discovery Performance Evaluation

In this section, we present the service discovery performance evaluation of TAO-DIS, discussing the obtained results in details. The objective of the evaluation of the discovery protocol TAO-DIS is to verify that the optimizations we applied to a fully epidemic protocol are effective. The dissemination of service descriptors should be as fast as with a fully epidemic protocol (considered as theoretically optimal with this respect), while reducing the load imposed on the network.

8.3.1 Simulation Setup

The series of simulations performed for the evaluation of TAO-DIS exploited artificial movements so that we could reach a greater number of involved devices (in our experiments up to 300 nodes). In an area of 1 km^2, a simple hybrid network was deployed that was composed of one infostation (in the center of the area) and a set of mobile devices carried by pedestrians (moving according to a random way-point mobility model, with a speed set to 0.5m/s). For each setup, we made 5 simulation runs with a different random seed. Furthermore, the number of service descriptors produced by the infostation varied between 1 and 3. We did not assign a lifetime and a maximum number of hops to the messages as these values are directly related to the nature of the provided services and their relative applications. Thus, the number of messages that roam the network increases continuously during the simulation. Each simulation ran during one hour before the infostations ceased to produce service descriptors. In these simulations we consider an infinite buffer size for both mobile devices and infostations.

8.3.2 Evaluation Metrics

For the evaluation of TAO-DIS, we mainly focused on the following two metrics:

- The *dissemination delay*. The dissemination delay is the time needed between the emission of a descriptor, and the time when all the clients have received the descriptor.

- The *network load*. It is the overall number of messages exchanged between the nodes at a specific instance of the simulation. Only messages carrying data related to the protocol were counted. Beacons emitted by the underlying neighboring discovery protocol, identical for the two compared protocols, were ignored.

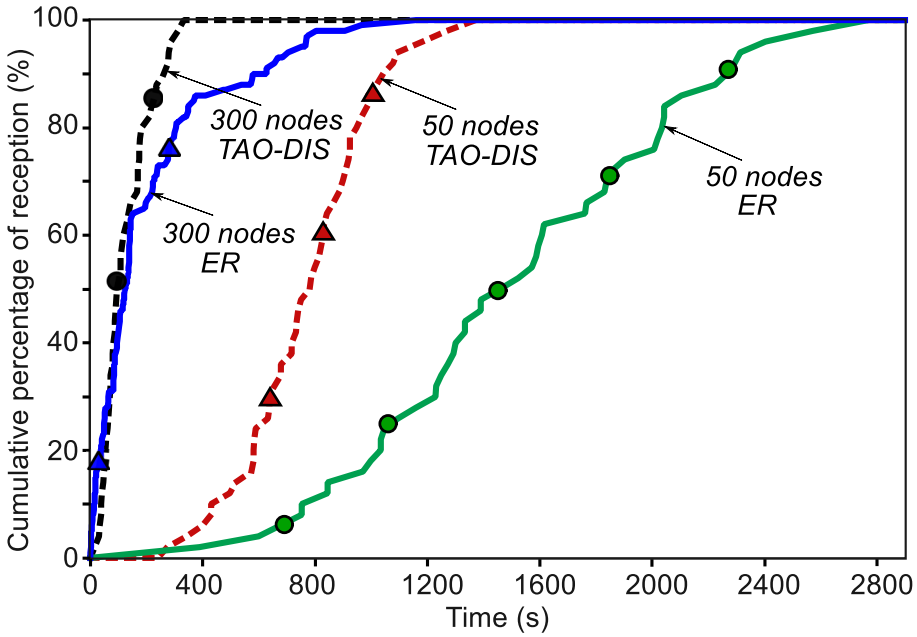


Figure 8.10: Performance evaluation of TAO-DIS: Average dissemination delay

8.3.2.1 Dissemination Delay

The first experiment was conducted to assess the delay for the dissemination of the service descriptors. As some descriptors may not arrive before the end of the simulation, we measure the cumulative percentage of reception along time, that is, at each instant, the proportion of clients that have received the descriptor (theoretically 100& after an infinite period of time). In the scenario considered, the infostation situated in the middle of the simulation area generated one descriptor (advertised in one SG in TAO-DIS) at the beginning of the simulation and we measured the time taken for this descriptor to reach the clients.

Figure 8.10 shows the results obtained with TAO-DIS, compared with a traditional fully epidemic protocol (the Epidemic Routing protocol [123]). The ER protocol disseminates service descriptors with a gossiping phase that first exchanges summary vectors containing the lists of known descriptors, in order to avoid sending duplicates of descriptors.

The curves display the cumulative percentage of clients that have received the descriptor along time, for two densities of devices (50 and 300 devices in the network, all of them being clients). TAO-DIS achieved its goal as it is never slower than the ER protocol. On the contrary, it can be seen that TAO-DIS overcomes the ER protocol, especially when the density of devices is low. For example, with 50 devices and after 1000 s, around 85% of the clients had received descriptors with TAO-DIS, whereas the proportion reach only less than 20% with the ER protocol. This is explained by the fact that the gossiping phase of the ER protocol is significantly longer than that of TAO-DIS, so this gossiping phase cannot successfully come to an end when the contact between neighbors is short (the de-

scriptor has no time to be transferred before the neighbors part). This has a greater impact when then density is low because the opportunities of contacts are scarcer and should be exploited at a maximum. This confirms the results presented in [103] that showed that the fraction of usable contacts decreases when the duration of discovery gossiping increases.

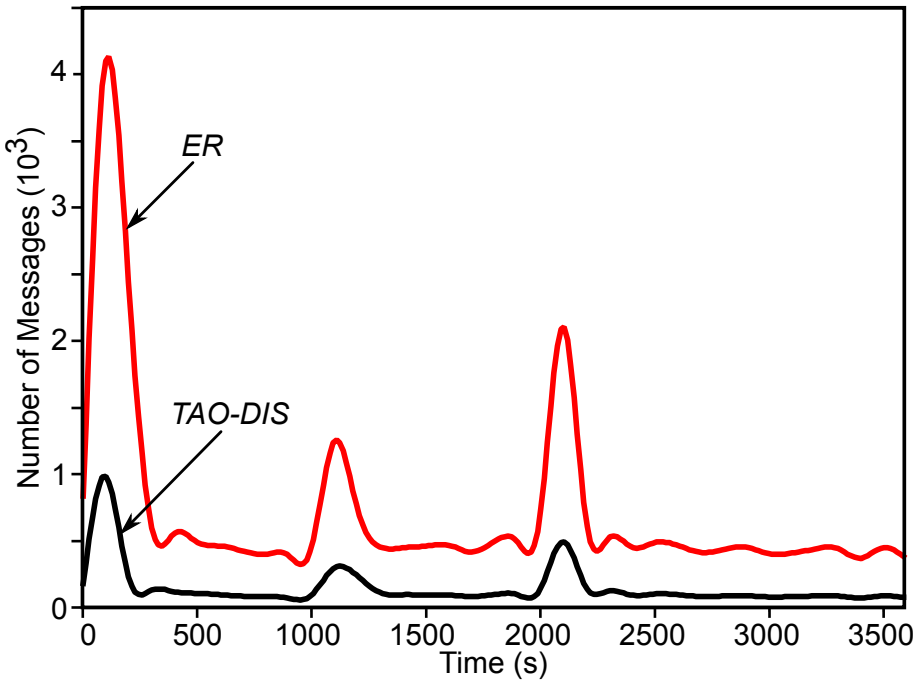


Figure 8.11: Comparison between TAO-DIS and a fully Epidemic dissemination protocol: Network Load

8.3.2.2 Network Load

In a second series of experiments, we compared again TAO-DIS with the ER protocol, focusing on the impact of our optimizations aiming at reducing the network load. Two metrics were actually considered: the number of messages sent and the amount of data sent. The infostation generated four different service descriptors during one-hour simulation period, that were eventually received by the 300 mobile devices. We introduce a 1000-second time gap between the generations of service descriptors (SGs in the case of TAO-DIS), scheduled at times $t = 0\ s$ (addition of two services), $t = 1000\ s$ (addition of one service) and $t = 2000\ s$ (addition of one service).

Figure 8.11 plots the evolution of the number of messages sent along time (a point x,y is plotted for y messages sent in the 50-second period ending at time x). AS can be noticed, the number of messages generated by the ER protocol is higher than that

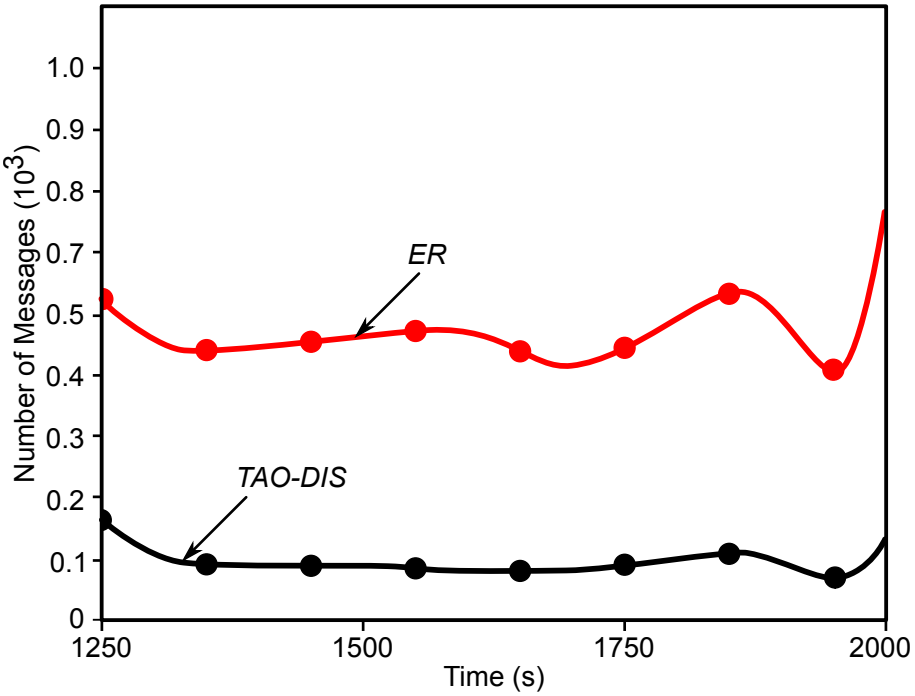


Figure 8.12: Close up comparison between TAO-DIS and a fully Epidemic dissemination protocol: Network Load

generated by TAO-DIS. A spike in the curve can be detected at the generation of a new service descriptor by the infostation. The short period that follows this generation shows an important difference between the two protocols. For example, between $t = 50\ s$ and $t = 100\ s$ around 420 messages are generated by the ER protocol versus 100 by TAO-DIS. This can be justified by the absence of summary vector exchanges between mobile devices in TAO-DIS. In addition, the grouping of service descriptors in one SG in the case of TAO-DIS reveals its interest when two simultaneous service additions are performed.

In Figure 8.12, a focus is made on the period during which the network is in a stable state, that is, when all the mobile devices have already been informed of al the available services and the infostation does not introduce any modification. One can clearly notice the difference between the two protocols. The number of exchanged messages by the ER protocol (around 450 messages every 50 seconds) reaches more than 5 times the exchanged by TAO-DIS (around 80 messages every 50 seconds). Indeed, TAO-DIS relies on only one exchange of a short offer message between two devices to check for the different SG versions. The ER protocol, on the other hand, still needs to periodically exchanged summary vectors at each contact to check for any updates of the provided services.

The difference between the ER protocol and TAO-DIS is similar when we consider the amount of data transferred, as shown in Figures 8.13 and 8.14. The use of hash keys and

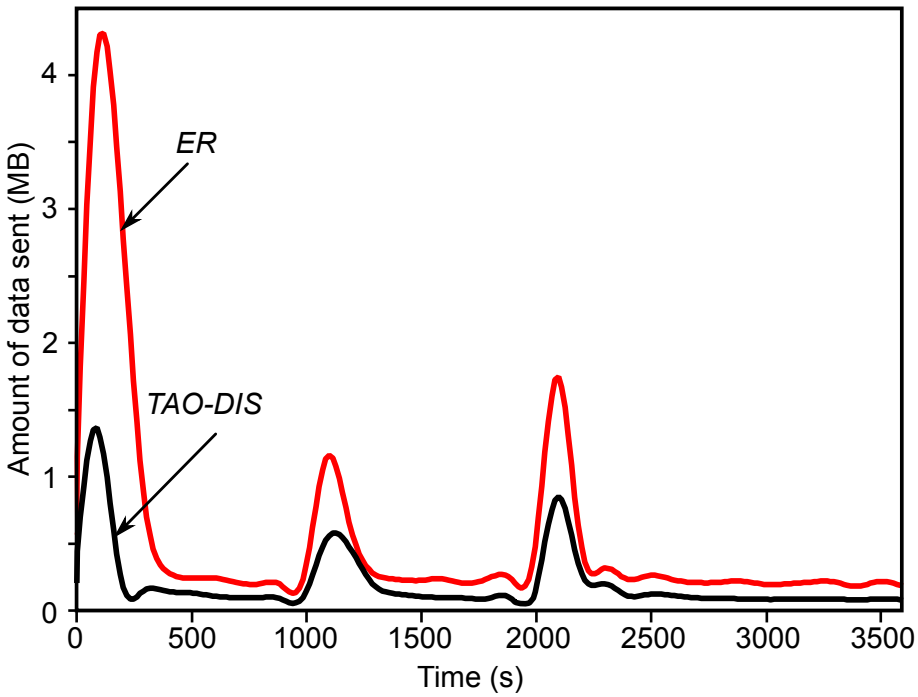


Figure 8.13: Comparison between TAO-DIS and a fully Epidemic dissemination protocol: Amount of exchanged data

the grouping of the descriptors in a single SG advertised by a simple hash key convey an evident advantage to TAO-DIS, particularly visible in the periods in which the network is stable.

8.4 Service Invocation Performance Evaluation

In this section, we present the service delivery performance evaluation of TAO-INV, discussing the obtained results in details. The general objective of these experiments is to measure the ability of TAO-INV to satisfy the client service delivery with a small number of message copies. To accomplish this objective we performed a set of comparisons with the state of the art protocols in order to evaluate specific parts of TAO-INV. In addition, we allowed ourselves to slightly modify these protocols in order to be fair when comparing with TAO-INV. These modifications will be detailed in the following section. Besides these comparison we performed another sets of experiments to tune the different parameters in TAO-INV.

8.4.1 Description of the Protocols Used for Performance Comparison

In the aim of evaluating the invocation process implemented in TAO (i.e., TAO-INV), we compared TAO-INV with three routing protocols RANDOM, Fresh and PRoPHET.

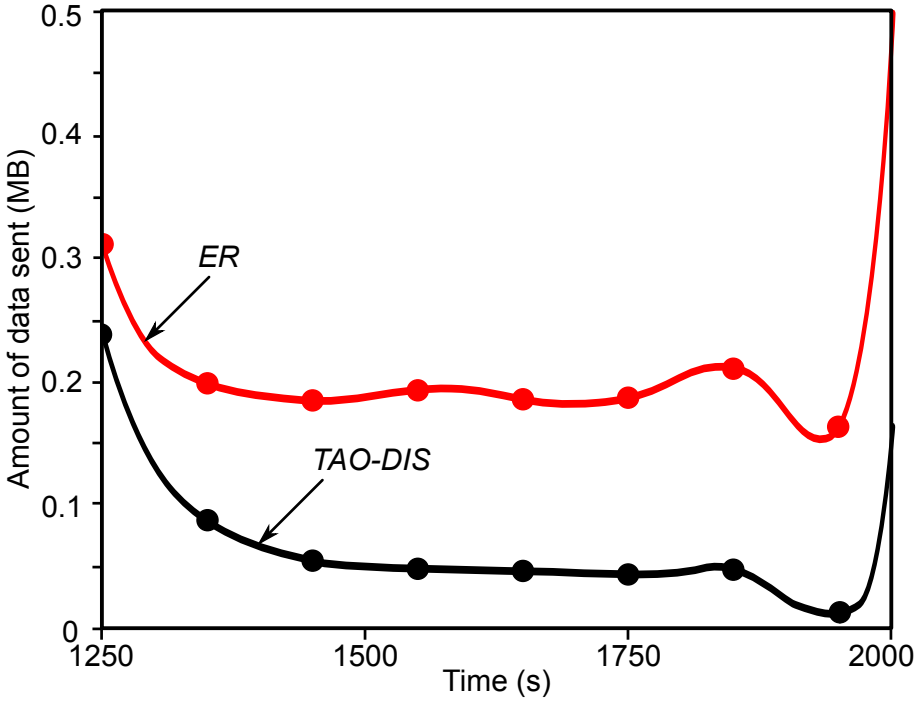


Figure 8.14: Close up comparison between TAO-DIS and a fully Epidemic dissemination protocol: Amount of exchanged data

RANDOM is a routing protocol that has relies on a random mechanism in choosing the next carriers of the service invocation messages among one-hop neighbors.

Fresh is a routing protocol that uses a similar key heuristic (the time heuristic) to perform its routing decisions [27]. Fresh targets connected MANETs, it builds a full end-to-end path between the source and the destination before initiating the sending of messages.

PRoPHET is a well known semi-epidemic protocol designed for intermittently connected mobile ad hoc networks [77]. It is not dedicated to service delivery but has some similarities with TAO-INV. In PRoPHET, the forwarding decisions are based on delivery probabilities estimated for destination nodes. These probabilities, based on the frequencies of encounters between pairs of nodes, are dynamically updated by each node when they encounter another one. A high delivery probability is assumed to reflect the fact that a node is a better carrier of a message towards its destination.

8.4.2 Evaluation Metrics

In our evaluation process of TAO, RANDOM, Fresh and PRoPHET, we focus on comparing their service delivery performance regarding the following metrics:

- The *satisfaction ratio*. It equals the number of invocations for which a client node receives a response from an infostation over the total number of invocations sent by clients. It reflects the ability of the protocol to eventually forward messages to their destination before the end of the simulation.

- The *average service delivery delay*. The service delivery delay is the delay between the emission of the service request by the client and the actual reception of the corresponding service response by the same client. Its theoretical lower bound is the one obtained when requests and responses are epidemically flooded in the network (with the assumption of the absence of interference and cache overflow). This service delay is averaged across all the requests/responses arrived before the end of the simulation time.
- The *network load*. It is the overall number of messages exchanged between nodes during the simulation. Only messages carrying data related to the protocols were counted. Beaconing messages dedicated to neighboring detection were discarded since we used the same beaconing frequency for all the protocols.

8.4.3 Comparison with RANDOM

The first set of experiments is the comparison of TAO-INV with the RANDOM protocol. The main objective of this set of simulations is not to compare the global performance of TAO-INV with RANDOM, but instead to assess the effectiveness of the time heuristic in delivering messages between a mobile node and a fixed infostation.

8.4.3.1 Protocols Parameters

Mobile clients are assumed to already have discovered the provided services and a client may invoke the desired service every 3 minutes.

- For *TAO-INV*: we have fixed the number of copies in the stock (E_{stock}) to 3 and the maximum number of emissions (E_{emit}) to 3. We have assigned low values to E_{emit} and E_{stock} in order to focus on the relevance and the choices done by the time-aware heuristic.
- For *RANDOM*: we have fixed the number of copies in the stock to 3.

8.4.3.2 Simulation Setup

The simulations performed for the evaluation of TAO-INV share the same setup as those detailed in sections 8.8 and 8.3.1. Where mobile nodes move at a speed equal to 0.5 m/s and clients may invoke the desired services every 3 minutes. Furthermore, the each simulation run is set to one hour with 10 minutes before and after the simulation for warm up and to allow the messages to be delivered. Finally, we consider an infinite buffer size for both mobile devices and infostations.

8.4.3.3 Results

First, we analyze the satisfaction ratio of each of the protocols in order to study the impact of increasing the number of nodes forming the network on the performance of each protocol. Three scenarios for each protocol presented in Figure 8.15, where each scenario

is characterized by the number of clients found in the network. As can be noticed, when having few nodes in the network, the satisfaction ratio of both protocols is almost the same. This observation is coherent with what is expected, because, due to the limited number of neighbors, TAO-INV and RANDOM will select most of the time the same carriers. The performance of TAO-INV increases with the number of nodes forming the network due to the selection of good carriers among a large set of neighbors. On the contrary, the performance of RANDOM decreases due to the bad selection of next carriers.

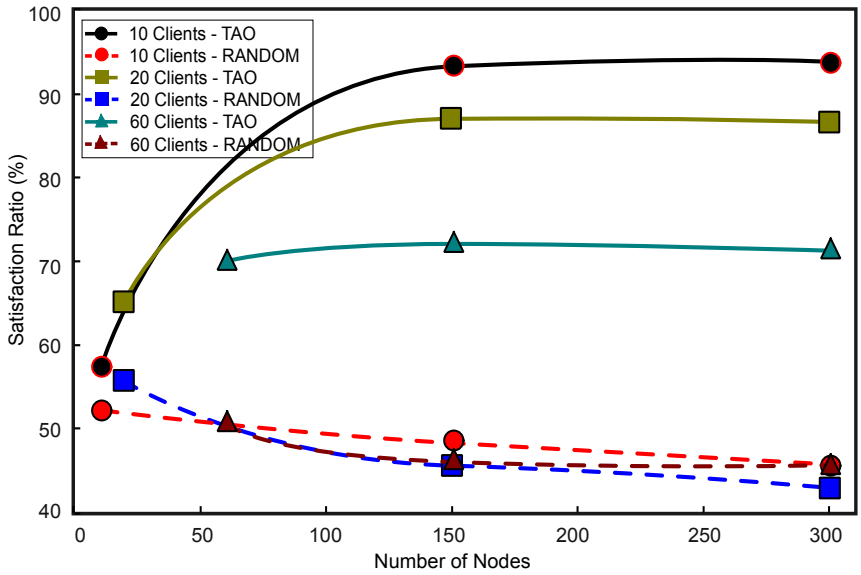


Figure 8.15: Comparison TAO-INV and RANDOM: Satisfaction ratio.

Based on Figure 8.16, we notice that when the network is formed of a few number of nodes, the delay values are relatively high. This is totally normal, since few nodes with limited transmission range and random waypoint mobility have to cover a large area. When the number of mobile nodes increases in the network, the average delay of RANDOM remains relatively high due to the random choices of carriers that contribute in transmitting the invocation messages toward the infostations. On the contrary, the average delay of TAO-INV decreases due to the presence of more carriers that can fill the gap between the client and the infostation and the ability of TAO in choosing good carriers to perform this operation.

Consequently, the obtained results show the effectiveness of the time heuristic implemented in TAO-INV in delivering messages between a mobile node and a fixed infostation while maintaining shorter delays.

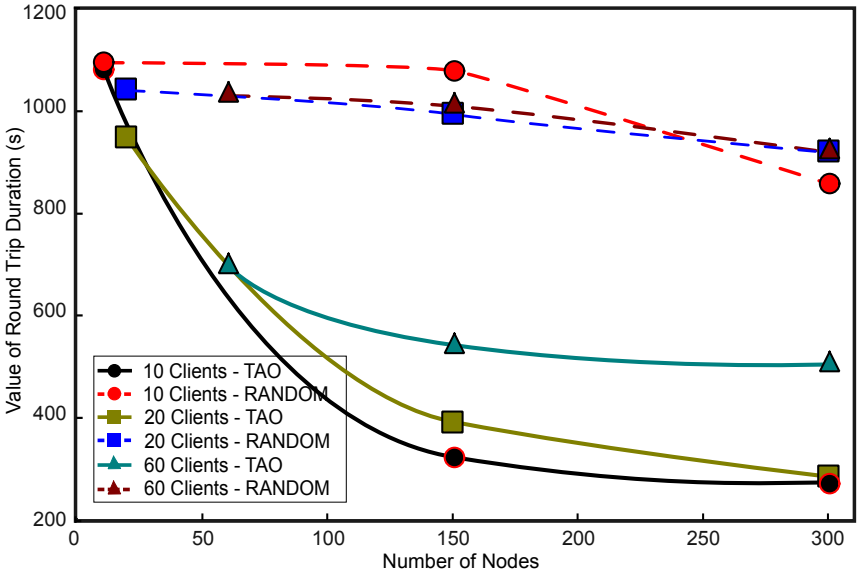


Figure 8.16: Delay of TAO and RANDOM routing protocols.

8.4.4 Comparison with Fresh

The main objective of this set of simulations is to show the importance of taking into account the type and properties of the targeted network in the designing phase of the protocol. Indeed, although Fresh and TAO-INV share a similar time heuristic, the obtained results show the difference in the performance between the two protocols. For this evaluation we perform two sets of experiments. The first set is a comparison with the original Fresh, it is the basic unaltered version of Fresh that is described by the authors in their paper [27]. In the second set, we perform the comparison with a modified version of Fresh we call it "ICMANET-compatible Fresh". In this version we introduce the "store, carry and forward" principle to Fresh so it can bare the disconnections that characterize ICMANETs.

8.4.4.1 Original Fresh

Fresh is designed to target connected MANETs. It builds a full end-to-end path between the source and the destination before initiating the sending of messages.

The simulation setups used for the experiments we conduct here are the same as those described in sections 8.8 and 8.3.1, and the parameters used for the protocols have the same values as those defined in section 8.4.3.1.

The metric used for the evaluation is the satisfaction ratio. It equals to the number of invocations for which a client node receives a response from an infostation over the total number of invocations sent by clients.

As expected, Fresh failed to function properly in ICMANETs. Table 8.5 shows that only a maximum of 2% of satisfaction ratio could be obtained while having 300 nodes

Number of Clients	Satisfaction Ratio
10	2%
20	2%
30	1.6%
40	1.8%
50	1.7%
60	1.8%

Table 8.5: Evaluation of Fresh while changing the number of clients (number of nodes = 300 node, radio range = 50 m)

roaming the network. These low values of satisfaction ratios are interpreted by the inability of Fresh to tolerate disconnections where as soon as a disconnection occurs along the path (between source and destination) the message is lost.

Even with very few disconnections Fresh performs poorly. This is verified with this series of experiments:

We changed the communication range of the nodes forming the network to reduce the frequency of disconnections in the network while changing the speed of the mobile nodes. Nevertheless, as shown in Tables 8.6a and 8.6b the maximum satisfaction ratio obtained is 9.2%. Indeed, if only one disconnection is encountered during the path building process from the source to the infostation (two way message exchange process) or during the transmission of the actual service request, the message is lost. This results in low satisfaction ratio values.

Range (m)	Satisfaction Ratio
50	1.8%
70	3.2%
100	8.8%

(a) Speed = 1.5 m/s

Range (m)	Satisfaction Ratio
50	2.26%
70	3.8%
100	9.2%

(b) Speed = 1 m/s

Table 8.6: Evaluation of Fresh while changing communication range and nodes' speed (number of nodes = 300, number of clients = 60)

8.4.4.2 ICMANET-compatible Fresh

In order to extend this comparison, we modified the behavior of Fresh to include the "store, carry and forward" concept. By that we converted a protocol originally designed to target MANETs to be able to overcome the frequent disconnections that characterize ICMANETs. Thus, when a disconnection is encountered the message is not lost but stored in the local cache of its forwarder till a forwarding opportunity appears. However, the algorithms implemented in Fresh concerning message forwarding and the choice of next carriers were unaltered.

As shown in Figure 8.17, the satisfaction ratio obtained by the ICMANET-compatible of Fresh is better than that obtained by the original one. Nevertheless, as the number of nodes forming the network increase, the satisfaction ratio decreases. This can be justified by the forwarding of a single copy of the service request. Indeed, Fresh is not designed to generate multiple copies of the service request. Unlike TAO-INV, Fresh does not perform a classification of the various neighbors as good or bad carriers, and as a result a message may end up forwarded to a carrier that may never encounter the message's final destination.

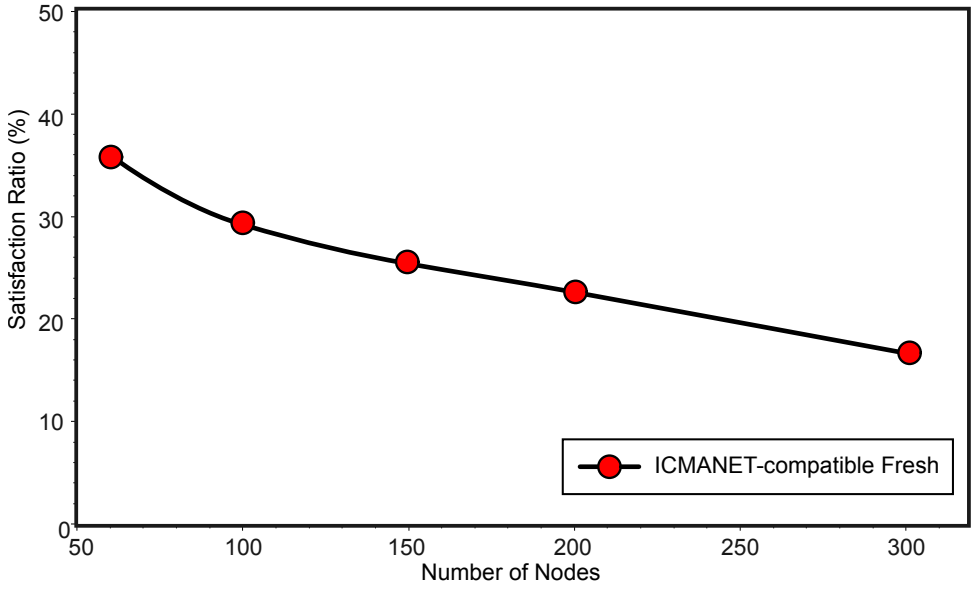


Figure 8.17: Satisfaction ratio of ICMANET-compatible Fresh (number of clients = 60, range = 50 m)

This suggests that it is not sufficient to implement the "store, carry and forward" concept over a routing protocol originally designed for MANETs to function efficiently in ICMANETs. This shows that the constraints and characteristics of the targeted network should be taken into consideration during the design phase of the routing protocol.

8.4.5 Comparison with PRoPHET

By comparing TAO-INV and PRoPHET, our main objective is to assess how TAO-INV can exploit the fact that the network is hybrid, i.e., to assess the effectiveness of a specialized heuristic based on the last time of contact with some fixed nodes (TAO-INV's heuristic), that makes sense only in an ICHN, compared to a somewhat more general heuristic based on the recurrence of contacts (PRoPHET's heuristic).

We chose PRoPHET for the comparison for several reasons: As it is fully specified in an RFC [76], it allowed us to implement it without making arbitrary choices that could impact the performances. PRoPHET is not dedicated to service delivery as it was not

designed to target the Request/Response communication pattern. Nevertheless, it has some similarities with TAO-INV. In PRoPHET, the forwarding decisions are based on delivery probabilities estimated for destination nodes. These probabilities, based on the frequencies of encounters between pair of nodes, are dynamically updated by each node when they encounter another one. A higher delivery probability is assumed to reflect the fact that a node is a better carrier of a message towards its destination.

8.4.5.1 Protocols Parameters

Similar to the other sets of experiments, mobile clients are assumed to already have discovered the provided services and a client may invoke the desired service every 3 minutes. The values of the various parameters of TAO-INV and PRoPHET are set to:

- For *TAO-INV*: we have fixed the number of copies in the stock (E_{stock}) to 3 and the maximum number of emissions (E_{emit}) to 3.
- For PRoPHET: we used the values of parameters suggested by the authors in their paper [77]. For that:
  - P_{init} was set to 0.75
  - B was set to 0.25
  - U was set to 0.98

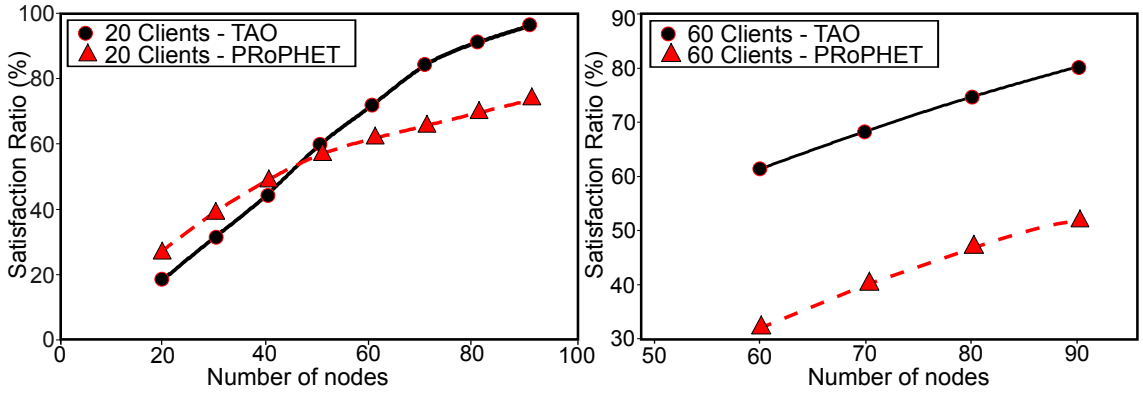
For this evaluation, we have performed experiments based on both the artificial and real movements already presented in section 8.2. Moreover, for the sake of comparison with the artificial environment, we added 3 infostations in the 4x4 km area, so that the infostations were distant from each other by more than 1 km. Again, roles of clients and relays were assigned to the mobile nodes.

8.4.5.2 Overall Results

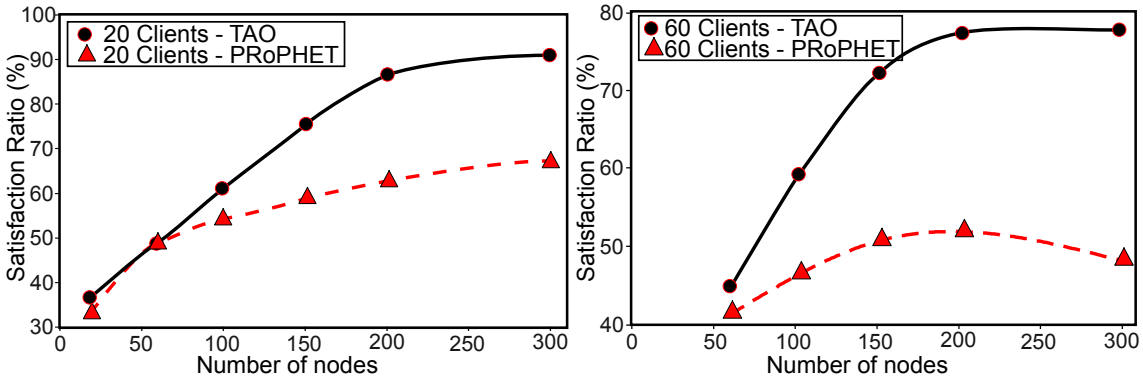
To perform the comparison between TAO-INV and PRoPHET we rely on the three metrics: satisfaction ratio, average service delivery delay and network load already defined in section 8.4.2.

Satisfaction Ratio Curves in Figure 8.18a (respectively 8.18b) present the variation of the satisfaction ratio of TAO-INV and PRoPHET when the number of mobile nodes in the network increases from 20 to 300 (respectively from 20 to 92), and the number of clients from 20 to 60 while moving according to the random way-point mobility model (RWP) (respectively real movement traces).

We can observe that, when having few nodes in the network, the satisfaction ratio provided by both protocols is almost the same. This observation is coherent with what



(a) Movements based on real traces

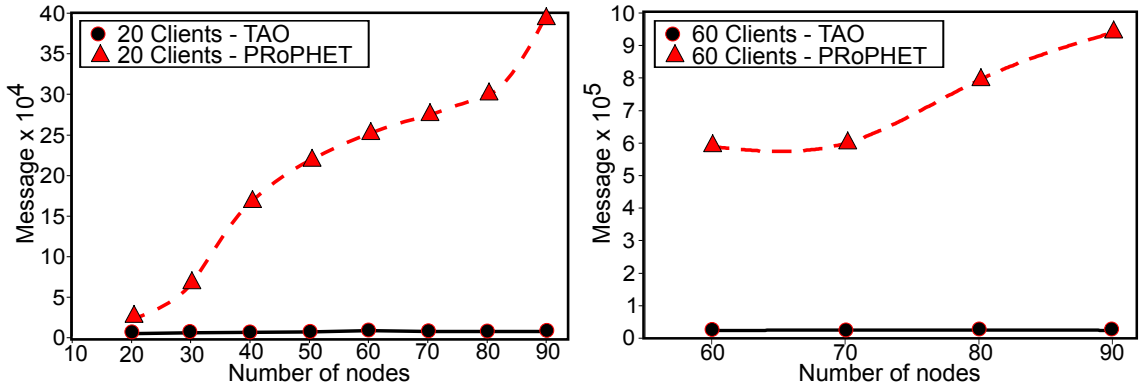


(b) Movements based on RWP

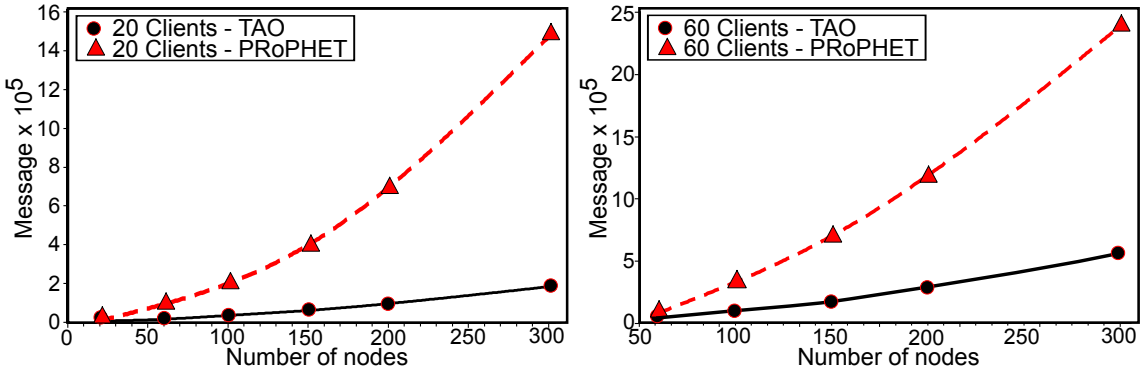
Figure 8.18: Comparison of TAO-INV and PRoPHET routing protocols: Satisfaction Ratio

is expected. Indeed when the network density is low, the contacts between mobile nodes themselves and with the infostation are occasional. In such a situation, TAO-INV and PRoPHET thus perform a quite similar selection of carriers. However, as the number of nodes forming the network increases, TAO-INV outperforms PRoPHET, especially when the proportion of clients among the nodes is high. With 60 clients in a network of 300 nodes, TAO-INV reaches a satisfaction ratio of 78% when PRoPHET is limited to 48%. Similarly, in the real traces case, TAO-INV reaches 81.3% of satisfaction ratio when PRoPHET is limited to 51% again under the same conditions.

Network Load Previous results are confirmed by those shown in Figures 8.19a and 8.19b. The network load induced by TAO-INV and PRoPHET is almost the same when the number of nodes forming the network is low. But, when this number increases, the network load increases in much larger portion with PRoPHET than with TAO-INV. In the RWP experiment, in which we managed to simulate up to 300 mobile nodes, the congestion level—that further analysis allowed us to situate at around 1 000 000 messages embedded into the network—is rapidly reached with PRoPHET.



(a) Movements based on real traces



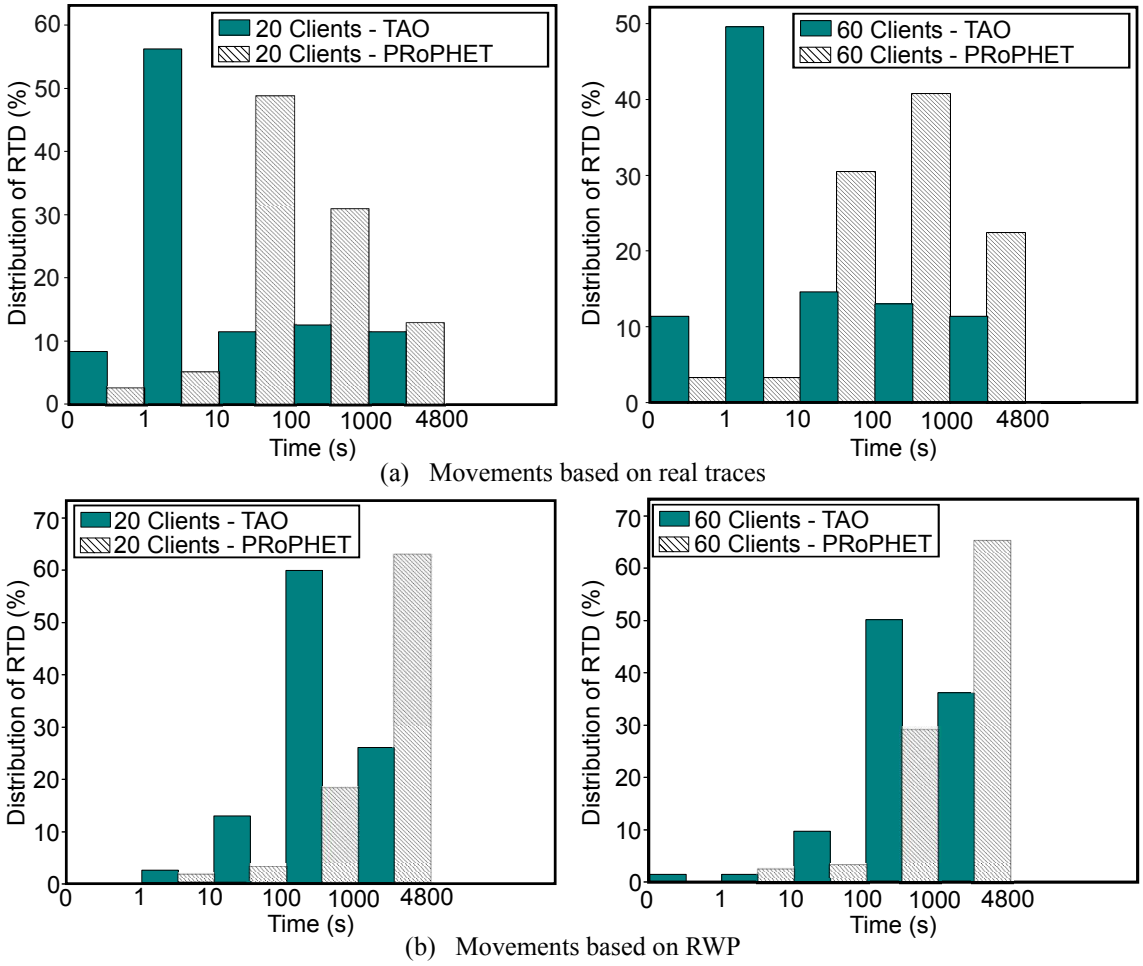
(b) Movements based on RWP

Figure 8.19: Comparison of TAO-INV and PRoPHET routing protocols: Network Load

The main explanation lies in the fact that sending a request with TAO-INV is more adapted to the hybrid characteristic of the network, as TAO-INV is specialized for sending requests only to infostations. Indeed, a TAO-INV node only embeds in the beaconing message information related to one node (the last date of contact with the infostation) whereas PRoPHET, in addition to beaconing, has to perform for each encountered node a gossip phase that consists in exchanging summary vectors related to every possible destination, PRoPHET being designed to be able to send a message to any host present in the network. Moreover, the timestamping-based heuristic implemented in TAO-INV permit somehow to reflect the notion of distance separating the mobile node from the infostation, whereas the frequency of contacts that PRoPHET relies on is unable to reflect this notion. Besides, the healing mechanism implemented in TAO-INV plays an important role in terminating the dissemination of service requests when the relative service response is received.

Service Delivery Delay Figures 8.20a and 8.20b represent the distribution of the round trip durations (RTD) for both TAO-INV and PRoPHET in the different scenarios we already mentioned. We notice that the majority of the services (more than 50%) need a duration of 100 to 1000 seconds for TAO-INV to be delivered in the RWP scenario, while more than 60% of the messages require a duration of 1000 to 4800 seconds for PRoPHET to be delivered. Similarly, in the real traces scenario, the majority of the round trip du-

rations fall in the 1 to 10 seconds period for TAO-INV, while it takes between 10 to 1000 seconds for PRoPHET.



(a) Movements based on real traces

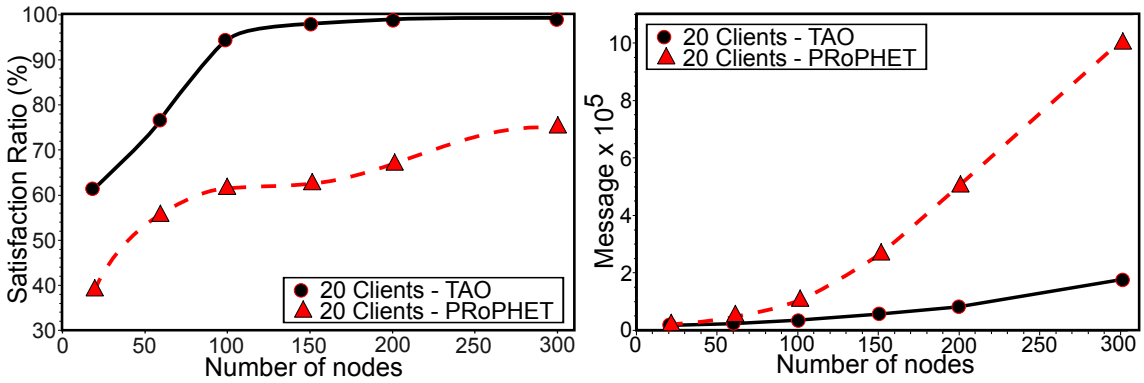
(b) Movements based on RWP

Figure 8.20: Comparison of TAO-INV and PRoPHET routing protocols: RTD

We notice that, in the case of RWP, the delay values are higher than that obtained in the real traces scenarios. This can be explained by the difference between these scenarios regarding the movements around the infostation. In the RWP scenario, the probability is not negligible that a node around the infostation, considered as good carrier by TAO-INV, changes suddenly its trajectory and rapidly recedes, eventually revealing itself unable to reach the infostation. This swift change of direction is far less probable in the real traces scenario as movements are constrained, typically by roads. On the other hand, lower delays are obtained by TAO-INV compared to those obtained by PRoPHET, this is due to the fact that PRoPHET requires the phase of summary vector exchange that mainly introduces a significant amount of delay to the message dissemination process. In addition, unlike TAO-INV, PRoPHET does not implement source routing and healing techniques to forward the service responses, thus increasing the gap between the two protocols when the number of nodes is high.

8.4.5.3 Impact of the Sole Timestamping Heuristic

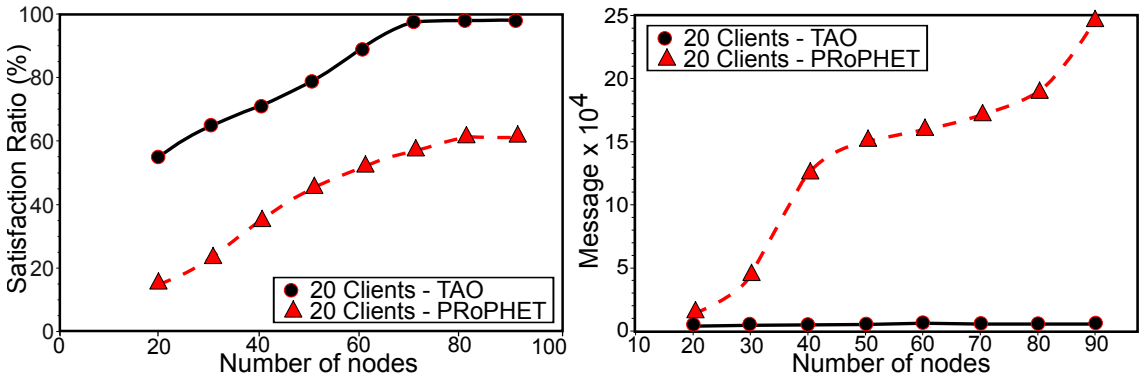
In order to evaluate the influence of the temporal heuristic on the overall service delivery performance of TAO-INV, we have conducted new simulations focusing only on the first phase of the invocation process, namely the forwarding of the service requests to the infostation, thus eliminating the impact of the source routing and the healing mechanisms implemented in TAO-INV. These evaluations were performed with the same parameters that had been used in the overall performance evaluations. We used the satisfaction ratio and the network load metrics with a modified semantics as we considered this time that an invocation was successful when the request had reached the infostation. Hence no responses were emitted.



(a) Satisfaction Ratio - RWP (b) Network Load - RWP

Figure 8.21: Evaluation of the timestamping-based heuristic: RWP

Figures 8.21 and 8.22 show the results obtained with each of the two metrics for the request-only invocation, along with a comparison to the performance of PRoPHET under the same conditions. Similar to the overall performance evaluations we performed these simulations using both the RWP and the real traces scenarios. As shown in Figures 8.21a and 8.22a, the overall performance of TAO-INV is clearly not only due to the use of source routing and healing as the satisfaction ratios obtained by TAO-INV (in both scenarios) are significantly greater than those obtained by PRoPHET. On the other hand, despite that Figures 8.21b and 8.22b clearly show that the number of messages disseminated in the first phase is almost equal to that in the overall invocation process of TAO-INV (i.e., the number of messages resulting from the source routing is negligible in comparison to the number of messages generated in the first phase), the load induced by TAO-INV on the network is still a lot lower than that induced by PRoPHET under the same conditions and the same mobility patterns.



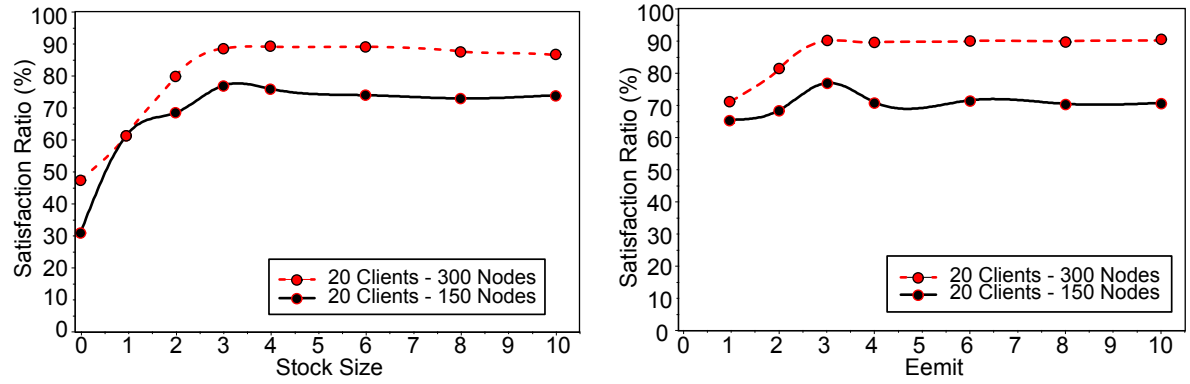
(a) Satisfaction Ratio - Real Traces (b) Network Load - Real Traces

Figure 8.22: Evaluation of the timestamping-based heuristic: Real Traces

8.4.6 TAO-INV Parameters Tuning

As we have introduced in Chapter 6 TAO-INV is a parametrized protocol, so it requires tuning. The two main parameters of TAO-INV are E_{emit} and E_{stock}. These parameters are directly related to the multi-copy message forwarding mechanism implemented in TAO-INV and should be controlled to ensure a high satisfaction ratio and a limited network load to avoid congestion. The main objective of this set of experiments is to see how TAO-INV reacts when tuning the different parameters.

For the simulation environment, we considered a 1 km^2 area with an infostation placed in the center of this area. The mobile nodes roam the network according to the RWP mobility. We fixed the number of client to 20 while varying the number of nodes from 150 to 300 nodes. Similar to all the other simulations, the obtained results are the average of 5 simulation runs with a different random seed.

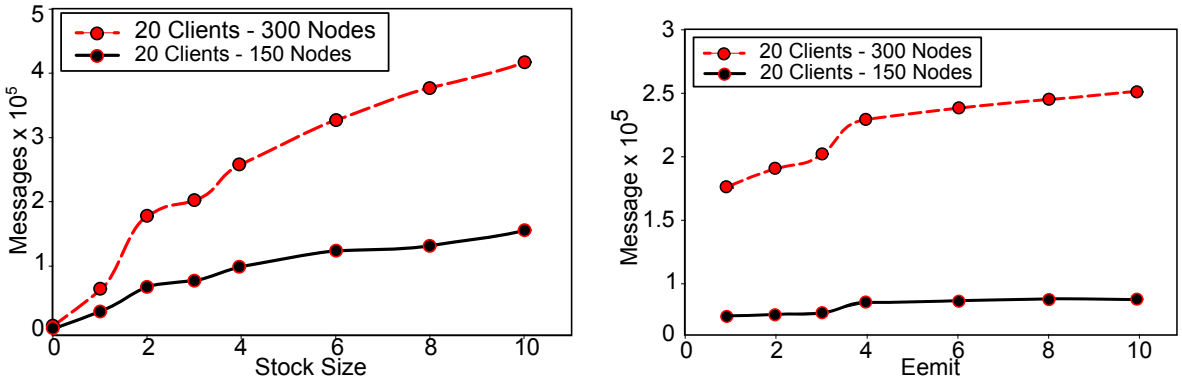


(a) Satisfaction Ratio while varying E_{stock} (b) Satisfaction Ratio while varying E_{emit}

Figure 8.23: Impact of E_{stock} and E_{emit} parameters on TAO-INV: Satisfaction Ratio

Figures 8.23a and 8.23b show the different values of the satisfaction ratio while tuning E_{stock} and E_{emit} respectively. We should note that, when tuning E_{emit}, $E_{stock}$is set to 3 and

vice versa. We clearly notice that when setting both E_{emit} and E_{stock} to 3, the maximum value of the satisfaction ratio is reached. When either of these parameters is set to a value greater than 3 we notice that there is no significant increase in the satisfaction ratio. This can be justified that the extra transmitted message copies are transmitted to bad neighbors that do not participate in delivering messages to their final destination.



(a) Network Load while varying E_{stock}

(b) Network Load while varying E_{emit}

Figure 8.24: Impact of E_{stock} and E_{emit} parameters on TAO-INV: Network Load

Figures 8.24a and 8.24b, on the other hand, represent the load imposed on the network as a result of varying these values. The curves show a significant increase in the number of generated messages when increasing the values of E_{emit} and E_{stock}. Nevertheless, the choice of the values of these two parameters should be a compromise between the satisfaction ratio and the network load. We notice that, choosing 3 as a fixed value for both parameters ensures getting the highest satisfaction ratio possible with the least possible number of generated messages.

8.5 Soft Handover Performance Evaluation

In this section, we present the simulation results we have obtained for the handover mechanism, and we analyze the impact of this mechanism on the service delivery from the client point of view. Similar to the previous set of simulations, these simulations have been performed on the OMNeT++ network simulator.

8.5.1 Environment

The environment we considered in these simulations is a square area of 1 $km \times 1$ km in which we have deployed 3 infostations (Figure 8.25). These infostations are connected together, and are placed 400 m away from each other. Each of them provides a specific

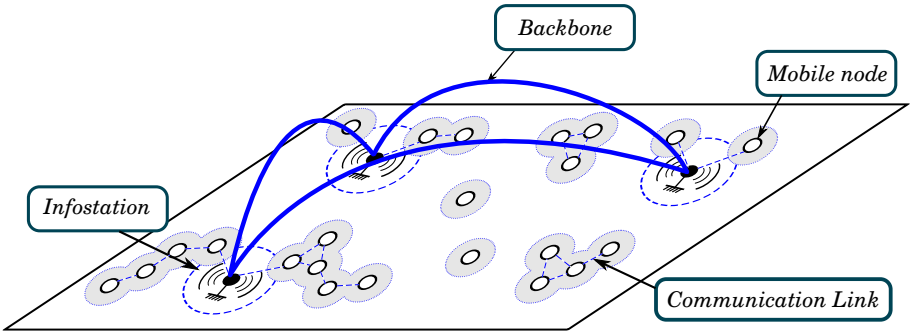


Figure 8.25: Simulation environment

service. These services are announced periodically (every 5 minutes) by all infostations. They can be discovered and invoked by pedestrians that move in this area using their handheld devices.

In these simulations, we consider two populations of pedestrians: the pedestrians that move following a random waypoint mobility model, and the pedestrians that move following predefined paths and that can exhibit their location. Both groups of the pedestrians move at a speed between 0.5 and 2 m/s.

In our simulations, 30 % of the mobile devices act as clients of the above-mentioned services, whereas the others only act as intermediate nodes. After discovering the services they are looking for, the clients invoke these services every 3 minutes. They are set to send a maximum of 10 requests during the simulations.

In our experiments, we have assigned to all the messages a lifetime of 10 minutes and a maximum number of hops of 10. The communication range of both mobile devices and infostations varies from 60 to 80 m.

Finally, we have considered successively 50, 100, 200 and 300 pedestrians in our simulations. All these parameters are defined so as to reflect as well as possible the behavior of humans that use their mobile phones when strolling in a city.

8.5.2 Evaluation Metrics

The objective of these experiments is to measure the impact of our handover solution on the service delivery in various configurations. For that, we focus especially on two values that reflect the quality of service that is perceived by the end-users (the ratio and the delay of service delivery), as well as on a value that shows the efficiency of the solution (the number of messages sent by all the nodes in the network throughout the whole simulation period).

We compare the performance of our solution with the Epidemic Routing protocol [123]. In Epidemic Routing, messages are flooded in the network and stored by all available neighbor nodes as a result of summary vector exchanges, thus maximizing the message delivery rate and minimizing message propagation latency. The first copy of a given service invocation request received by an infostation (or the first copy of a given service response received by a client) has therefore followed the path that offers the shortest delivery delay. Moreover, since the responses are disseminated by all the nodes, including

the infostations, no handover mechanism is required with this protocol. In this context, the Epidemic Routing protocol appears as a good candidate to evaluate the efficiency of our solution, even if no precautions are taken in this protocol to limit the number of messages that are disseminated.

8.5.3 Comparison with Epidemic Routing Protocol

Figures 8.26, 8.27 and 8.28 represent the simulation results we have obtained. We can observe that our solution offers a better service delivery in terms of ratio and delay than the epidemic routing protocol, while reducing drastically the number of messages that are forwarded in the network, especially when the number of nodes increases.

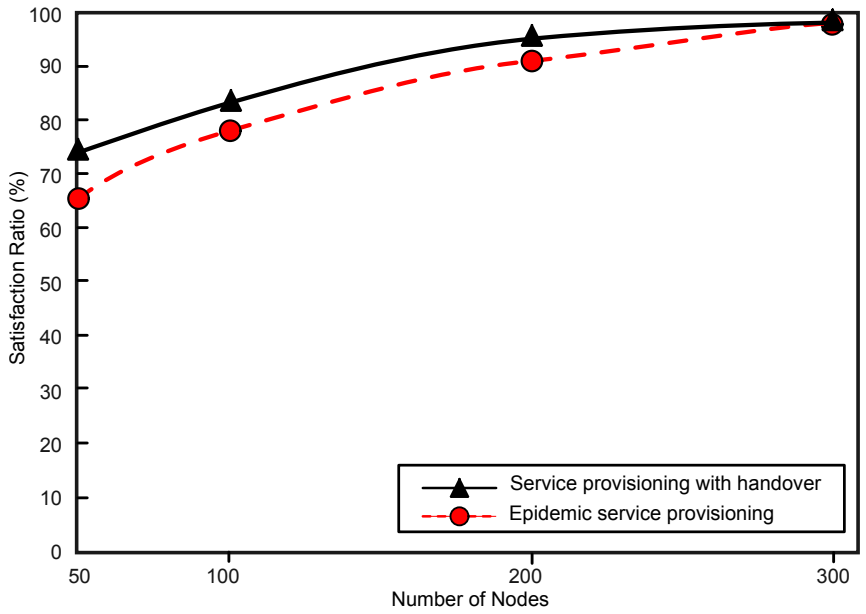


Figure 8.26: Comparison between Soft Handover and Epidemic Routing protocol: Satisfaction Ratio

Curves in Figure 8.26 show the variation of the satisfaction ratio of the soft handover mechanism we are proposing and the Epidemic service provisioning when the number of mobile nodes in the network increases from 50 to 300. We can observe that, the satisfaction ratio of our proposition is higher than that obtained by the Epidemic Routing protocol. However, for both protocols when having few nodes in the network the satisfaction ratio is low (65 % for Epidemic and 74 % for handover) compared to the values obtained when having more nodes in the network (reaching 98 % for both protocols). This observation is coherent with what is expected, because more good carriers can be found among a large set of neighbors, thus reducing the number of disruptions and the disconnection times in the routes.

As for the curves in Figure 8.27, they show the average service delivery delay values obtained by the two protocols. We notice that, the difference is more observable when the

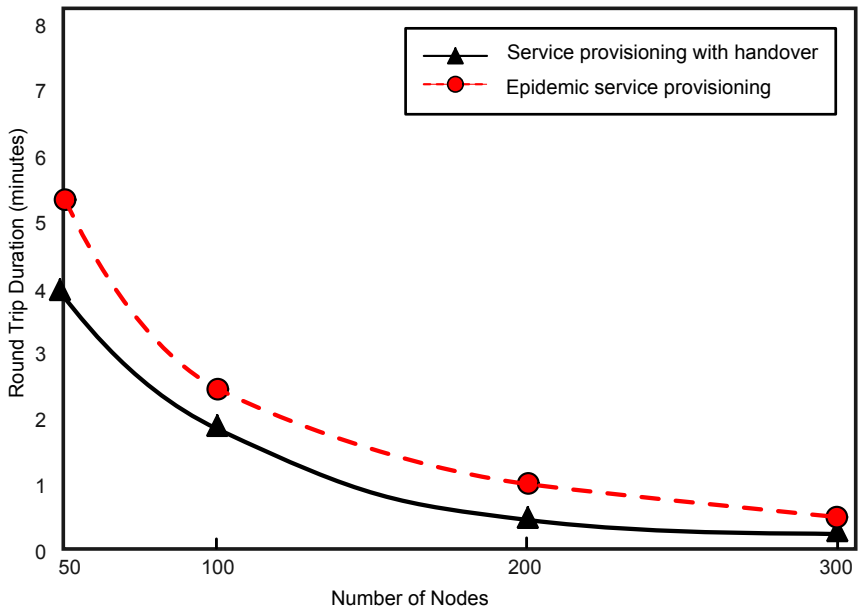


Figure 8.27: Comparison between Soft Handover and Epidemic Routing protocol: Service Delivery Delay

number of devices is low (4 minutes for the handover and 5.2 minutes for the epidemic) because it is more difficult to find another intermediate node. Moreover, the delivery delays and the satisfaction ratios are often better with our proposal because the message forwarding is always coupled with the handover mechanism resulting in the intervention of the infostation closest to the client, while with the epidemic routing protocol the messages are forwarded after the summary vector exchanges. Due to this short additional latency in the message forwarding, some communication disruptions can occur in certain situations, thus reducing the opportunities to forward the messages.

Finally, the results represented in Figure 8.28 show an important difference between the handover and the Epidemic Routing protocol. The number of messages exchanged by the handover protocol reaches around 2000 messages when having 300 mobile nodes in the network, while the Epidemic protocol bypasses this value to reach around 4500 exchanged messages. The handover mechanism relies on benefiting from the closest infostation to the client, thus reducing the number of exchanged messages. On the other hand, the Epidemic Routing protocol relies on the exchange of summary vectors that introduces a large load on the network.

8.6 Conclusion

In the first section of this chapter we have presented the general architecture of the TAO platform and the implementation details including the API. The next section covered the performance evaluation of the three protocols we have introduced (TAO-DIS, TAO-INV and soft handover). These evaluations include comparing the performance of the proposed protocols with state of the art protocols and tuning the various parameters. The obtained results confirm the validity of our propositions in the majority of the performed

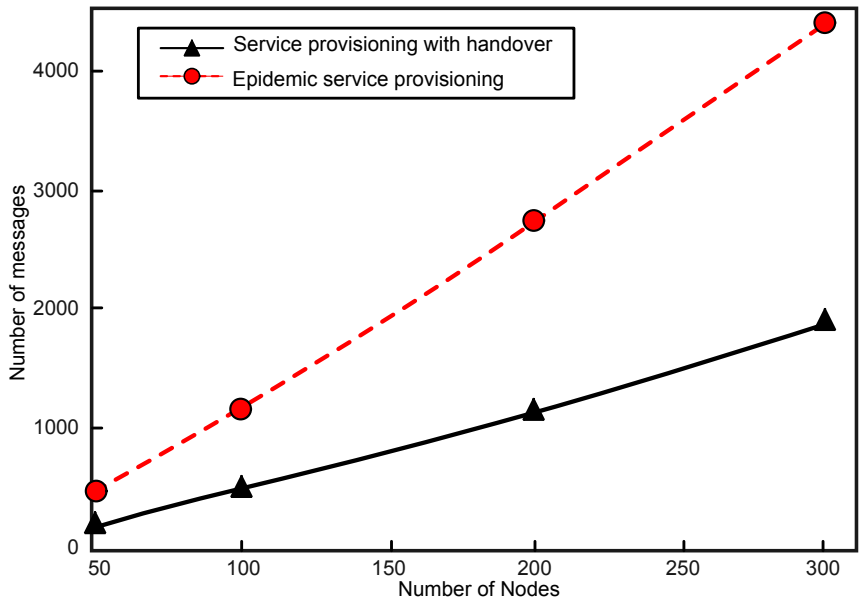


Figure 8.28: Comparison between Soft Handover and Epidemic Routing protocol: Network Load

experiments. We aimed to generalize the results by relying on both simulated and real traces for the mobile nodes' movements.

Concerning TAO-DIS, the simulations show that our proposition is able to match epidemic dissemination delay wise while noticeably reducing the number of messages and the amount of exchanged data in comparison with the fully Epidemic dissemination protocol.

As for the TAO-INV simulations, obtained results show that when designing a routing protocol it is important to take into account the nature of the targeted network. For that, TAO-INV was able to outperform protocols originally designed to target MANETs (comparison with Fresh) even after implementing the "store, carry and forward" principle in Fresh. Furthermore, TAO-INV was able to outperform a general purpose routing protocol (comparison with PRoPHET), by that showing the importance of designing a routing protocol that takes into account the hybrid nature of the network.

Concerning the soft handover mechanism, the obtained results show that the soft handover mechanism we are proposing is able to ensure better delay, satisfaction ratio and network load in comparison with an Epidemic Routing protocol. Indeed, such results are obtained since we take into consideration the possibility of benefiting from the closest infostation to the client thus relying on faster and closer paths toward the final destination of the messages.

9 Conclusions and Future Work

The main objective of this thesis was presented in Chapter 1 and was summarized by exploiting the specific characteristics of ICHNs to optimize the pervasive service provisioning in an opportunistic way. Throughout the document we presented the various constraints we faced due to the challenged network type we are dealing with and the solutions we proposed. In this chapter, we discuss how we have fulfilled this objective and we present some future lines of research to follow.

9.1 Summary of Contributions

In this thesis, we have tackled the service provisioning challenge in pervasive computing environments. Our objective was to exploit the specific characteristics of ICHNs in order to optimize the pervasive service provisioning in an opportunistic way. For that, we proposed TAO, a Time-Aware Opportunistic platform to perform service provision in ICHNs. TAO is composed of two protocols and implements a handover mechanism:

- *TAO-DIS protocol*: it is the protocol related to the service discovery phase of the service provisioning challenge in ICHN.
- *TAO-INV protocol*: it is the protocol related to the service invocation phase of the service provisioning challenge in ICHN.
- *Soft handover mechanism*: an optimization over invocation phase, introduced to achieve better service delivery for clients.

The research work developed in this thesis starts by introducing a discovery protocol that insures a rapid, light and automatic discovery of services in the network. Fulfilling the objective of informing all potential clients about the offered services in the network, TAO-DIS ensures a fast dissemination that matches an epidemic one while limiting both the number and the size of exchanged messages in the network. TAO-DIS is able to reach such performance through the merging of service descriptors in what we call a service guide and relying on a hash function to generate distributed IDs to both service descriptors and service guides. Thus, it is sufficient to exchange hash keys to detect any change in the different service guides stored in the local caches of each mobile nodes. This approach was constructed and compared with the Epidemic Routing protocol. The results of these tests indicated that the ideas presented in TAO-DIS noticeably reduce the

number and the size of exchanged messages while ensuring a full dissemination of the SGs to all mobile devices roaming the network.

TAO-INV is our second contribution to service provisioning in ICHNs. The inability of current routing protocols to effectively deal with ICHNs led us to the design of a routing protocol that takes into consideration the hybrid nature of the network when designing a routing protocol. Indeed, TAO-INV takes into account this hybrid nature as it relies on a timestamping-based heuristic to estimate the ability of intermediary nodes to deliver a message to an infostation. This heuristic is computed depending on the last contact time between a mobile node and an infostation. This utility-based forwarding epidemic is carefully mixed with counter-based rules that aim at reducing the number of copies spread in the network. We have validated our protocol using both real traces and artificial movements, showing that it is able to guarantee good performance. When compared to other routing protocols (general purpose protocols such as PRoPHET), TAO-INV has remarkable differences: the small amount of information needed to perform the routing decisions, the low number of messages generated and the high satisfaction ratio achieved.

In order to optimize the interaction of infostations with mobile clients, we introduced the soft handover mechanism that aims at choosing the best infostation among a set of infostations belonging to the same cluster to forward a service response back to a specific client. This handover mechanism relies on the unlimited pair-wise contacts among mobile devices and infostations. By that, estimations of the quality of the discontinuous multi-hop paths between infostations and clients are performed. These estimations are performed in the infostations and rely on the stability and the length of the paths. Upon reception of a service request, infostations of the same cluster elect, based on the estimation values, the best infostation to forward the service response to the client. Comparing our handover mechanism with the Epidemic Routing protocol showed that our proposition is able to ensure better performance in terms of service delivery, network load and even service delivery delay.

Throughout this document, we have presented the design, evaluation and implementation of our proposed solutions TAO-DIS, TAO-INV and the soft handover mechanism, all specifically designed for ICHNs.

9.2 Heading Beyond

We conclude this thesis with the identification and discussion of some open research issues. The following list we present is, of course, not exhaustive; instead, it intends to underline some important points of a research agenda for future work. There are many other interesting issues that remain to be resolved or improved in our proposals, such as security or robustness.

The results and solutions presented in this work can represent a foundation for a wide research agenda in the area of intermittently connected hybrid networks and, more in general, of mobile systems. Indeed, this work represents the first steps toward achieving adaptive middleware for resource-constrained ICHNs. Clearly, additional application case studies can be run to better evaluate the proposed middleware systems.

Concerning TAO-DIS
A key contribution of TAO-DIS is the full dissemination of provided services on all mobile clients in the network. The current implementation achieves this objective with minimum overhead and amount of data exchanged. Indeed, with TAO-DIS we introduce an optimization on the fully Epidemic dissemination protocol to ensure having the same dissemination delay. While this is sufficient to enable the discovery of provided services, there will certainly be extra research in this area.

In the current specification of TAO-DIS, semantics are not taken into account. This approach is not new to the context of ad hoc networks. For example, several research works **[137, 58, 110]** have relied on various languages, such as OWL-S, JSON, etc to describe services and to achieve service discovery. Furthermore, other approaches rely on context information (e.g., location, device profile, environment parameters) to perform service discovery to find services that best fit a particular context [4, 67]. We believe that, studying the effect of limiting the content of SGs to the interests of clients can reduce the size of the SGs but, on the other hand, it has a major impact on the dissemination process.

Another area of research is service matching. In the current version of the middleware, the service matching process is performed manually by the client. After the discovery process, a client manually chooses the desired service and invokes it. In our opinion, a service matching mechanism should be implemented in order to automatically invoke services needed by applications on the device, where no manual intervention from the user should be needed. Actually, service matching is associated with the semantics and language used for the service description.

Concerning TAO-INV
Regarding TAO-INV, the key contribution is permitting mobile clients to invoke remote services found on infostations and receive the response back despite their mobility and the intermittent disconnections in the network. In the current implementation of TAO-INV forwarding decisions are strictly based over the timestamping-based utility where no location information (such as: GPS information, GSM tower triangulation, Wi-Fi Triangulation, etc.) are utilized. We argue that improvements on TAO-INV can still be obtained by relying on location information such as GPS when available. Thus, a node can choose next carriers by relying on their speed and direction with respect to the position of the infostation providing the service. This should mainly reduce the overhead and the overall delay.

Concerning the soft handover mechanism
The soft handover mechanism we proposed aims at choosing the best infostation to send a service response back to a specific client according to the opportunistic nature of the communications in the network. In our proposition infostations do not benefit from the presence of other clusters in the network. We argue that creating a type of communication between different clusters might help in improving the performance of service delivery and in turn access continuity as different clusters are more likely distributed over wider physical area. This will permit reaching mobile clients faster even if no infostations of the same cluster are deployed there.

Real world experiments
Due to limitations in the number of devices that can be deployed in real world experiments, we relied in our evaluations on simulations experiments. In these experiments we utilized both real and artificial movements. Nevertheless, for future work we should conduct a series of real world environment experiments to perform a more detailed evaluation of the performance of the service provisioning middleware we are proposing. The next step for this work is to make the source code for the TAO-DIS, TAO-INV and the soft handover mechanism available to the community coupled with a proper documentation.

From general purpose to specialized protocols
Finally, in this thesis, we have started a lead of designing specialized protocols for ICHNs. We argue that avoiding general purpose solutions for specialized problems is a good approach to follow. Indeed, by comparing our protocols to general purpose protocols we have observed that specialized solutions deal better with the various constraints and achieve better performance. We believe that, an interesting research topic would be considering this approach of designing specialized protocols to solve other challenges in intermittently connected networks. For example, one of these challenges could be tuple space in intermittently connected networks. Despite the frequent disconnections, the aim is to have a shared data space acting as an associative memory used by the various devices forming the network [5]. Another challenge that can be considered is the consensus problem, where multiple processes running on different mobile devices must reach an agreement of a common course of actions [6]. The idea of designing specialized protocols and algorithms to tackle such challenges could be interesting and more efficient if they would be specifically designed for chosen scenarios or case studies instead of relying on general purpose protocols that might not adapt to the challenging characteristics of ICMANETs and ICHNs.

Bibliography

[1] DTNRG, Delay Tolerant Networking Research Group, IRTF Internet Research Task Force. www.dtnrg.org.

[2] *Universal Description Discovery and Integration Platform*, September 2000.

[3] Daniel Aguayo, John Bicket, Sanjit Biswas, and Douglas De Couto. MIT Roofnet: Construction of a Production Quality Ad-Hoc Network, September 2003.

[4] Eyhab Al-Masri and Qusay H. Mahmoud. A Context-aware Mobile Service Discovery and Selection Mechanism Using Artificial Neural Networks. In *Proceedings of the 8th International Conference on Electronic Commerce*, ICEC '06, pages 594–598, Fredericton, New Brunswick, Canada, 2006. ACM.

[5] Abdulkader Benchi, Pascale Launay, and Frédéric Guidec. A P2P Tuple Space Implementation for Disconnected MANETs. *Peer-to-Peer Networking and Applications*, pages 1–16, August 2013.

[6] Abdulkader Benchi, Launay Pascale, and Frédéric Guidec. Solving Consensus in Opportunistic Networks. In *Proceedings of the 16th International Conference on Distributed Computing and Networking*, ICDCN '15, Goa Campus, India, January 2015. ACM.

[7] Micha Benoliel, Stanislav Shalunov, and Greg Hazel. Open Garden Project.

[8] Guy Bieber and Jeff Crpenter. Introduction to Service-Oriented Programming. In *OpenWings Whitepaper*, April 2001.

[9] Chiara Boldrini, Marco Conti, Jacopo Jacopini, and Andrea Passarella. HiBOp: a History Based Routing Protocol for Opportunistic Networks. In *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks*, WoWMoM '07, pages 1–12, Helsinky, Finland, 2007. IEEE Computer Society.

[10] Chiara Boldrini, Marco Conti, and Andrea Passarella. Exploiting Users Social Relations to Forward Data in Opportunistic Networks: the HiBOp Solution. *Pervasive and Mobile Computing (PMC)*, 4(5):633–657, 2008.

[11] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proceedings of the 25th IEEE Conference on Computer Communications*, INFOCOM '06, Barcelona, Catalunya, Spain, April 2006. IEEE Communications Society.

[12] Celeste Campo, Mario Munoz, Jose Carlos Perea, Andreas Marin, and Carlos Garca-Rubio. PDP and GSDL: A New Service Discovery Middleware to Support Spontaneous Interactions in Pervasive Systems. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications Workshops*, PerCom '05.

[13] Antonio Carzaniga, Matthew J. Rutherford, and Alexander L. Wolf. A Routing Scheme for Content-Based Networking. In *Proceedings of the 23rd IEEE Conference on Computer Communications*, volume 2 of *INFOCOM '04*, pages 918–928, Hong Kong, China, March 2004. IEEE.

[14] Antonio Carzaniga and Alexander L. Wolf. Content-based Networking: A New Communication Infrastructure. In *Proceedings of the National Science Foundation Workshop on an Infrastructure for Mobile and Wireless Systems*, NSF '01, pages 59–68, Scottsdale, Arizona, October 2001. Springer-Verlag.

[15] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Pocket Switched Networks: Real-world Mobility and Its Consequences for Opportunistic Forwarding. Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Laboratory, February 2005.

[16] Dipanjan Chakraborty, Anupam Joshi, and Yelena Yesha. Integrating service discovery with routing and session management for ad-hoc networks. *Ad Hoc Networks*, 4(2):204–224, March 2006.

[17] Dipanjan Chakraborty, Anupam Joshi, Yelena Yesha, and Tim Finin. Toward Distributed Service Discovery in Pervasive Computing Environments. *IEEE Transactions on Mobile Computing*, 5(2):97–112, February 2006.

[18] Thomas Clausen, Philippe Jacquet, Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, Laurent Viennot, et al. Optimized Link-State Routing Protocol (OLSR). Technical report, Project Hipercom, INRIA, October 2003.

[19] Salutation Consortium. Salutation Architecture Specification, 1999.

[20] Marco Conti, Silvia Giordano, Martin May, and Andrea Passarella. From Opportunistic Networks to Opportunistic Computing. *IEEE Communications Magazine*, 48(9):126–139, September 2010.

[21] Marco Conti and Mohan Kumar. Opportunities in Opportunistic Computing. *Computer*, 43:42–50, 2010.

[22] Microsoft Corporation. Universal Plug and Play: Background, 1999.

[23] Paola Costa, Mirco Musolesi, Cecilia Mascolo, and Gian Petro Picco. Adaptive Content-based Routing for Delay-tolerant Mobile Ad Hoc Networks. Technical report, UCL, August 2006.

[24] Paolo Costa, Cecilia Mascolo, Mirco Musolesi, and Gian Pietro Picco. Socially-aware Routing for Publish-subscribe in Delay-tolerant Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 26(5):748–760, June 2008.

[25] Elizabeth M. Daly and Mads Haahr. Social Network Analysis for Routing in Disconnected Delay-tolerant MANETs. In *Proceedings of the 8th International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '07, pages 32–40, Montreal, Quebec, Canada, 2007. ACM.

[26] Avri Doria, Maria Uden, and Durga P. Pandey. Providing Connectivity to the Saami Nomadic Community. In *Proceedings of the 2nd International Conference on Open Collaborative Design for Sustainable Innovation*, Bangalore, India, December 2002.

[27] Henri Dubois-Ferriere, Matthias Grossglauser, and Martin Vetterli. Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages. In *Proceedings of the 4th International Symposium on Mobile Ad Hoc Networking and Computing*, Mobihoc '03, pages 257–266, Annapolis, MA, USA, 2003. ACM.

[28] Thomas Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.

[29] Kevin Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *Proceedings of the Special Interest Group on Data Communication*, SIGCOMM '03. ACM, August 2003.

[30] Kevin Fall and Stephen Farrell. DTN: An Architectural Retrospective. *IEEE Journal on Selected Areas in Communications*, 26(5):828–836, June 2008.

[31] Kevin Fall, Wei Hong, and Samuel Madden. Custody Transfer for Reliable Delivery in Delay Tolerant Networks. Technical report, Intel Research Berkeley, 2003.

[32] Richard Frenkiel and Tomasz Imieliński. Infostations: The joy of many-time many-where communications. *Journal on Mobile Computing*, May 1996.

[33] Zhenguo Gao, Ling Wang, Mei Yang, and Xiaozong Yang. CNPGSDP: An efficient group-based service discovery protocol for MANETs. *Computer Networks*, 50(16):3165–3182, November 2006.

[34] Paul Gardner-Stephen. The Serval Project: Practical Wireless Ad-hoc Mobile Telecommunications. *Rural, Remote & Humanitarian Telecommunications Fellow*.

[35] David Goodman, Joan Borras, Narayan B. Mandayam, and Roy Yates. Infostations: A New System Model for Data and Messaging Services. In *Proceedings of the 47th Vehicular Technology Conference*, VTC '97, pages 969–973, Phoenix, AZ, USA, May 1997. IEEE.

[36] Matthias Grossglauser and David N.C. Tse. Mobility Increases the Capacity of Ad hoc Wireless Networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, Aug 2002.

[37] Zhen guo Gao, Xiao zong Yang, Tian yi Ma, and Shao bin Cai. RICFFP: An Efficient Service Discovery Protocol for MANETs. In *Embedded and Ubiquitous Computing*, Lecture Notes in Computer Science, pages 786–795. Springer Berlin / Heidelberg, 2004.

[38] Piyush Gupta and Panganmala R. Kumar. The Capacity of Wireless Networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, Mar 2000.

[39] Erik Guttman, Charles Perkins, John Veizades, and Michael Day. Service Location Protocol, Version 2. IETF RFC 2608, 1999.

[40] Zygmunt J. Haas, Marc R. Pearlman, and Prince Samar. The Zone Routing Protocol (ZRP) for Ad Hoc Networks. Internet-draft, IETF MANET Working Group, July 2002.

[41] Julien Haillot and Frédéric Guidec. Towards a Usenet-like Discussion System for Users of Disconnected MANETs. In *Proceedings of the 1st IEEE International Workshop on Opportunistic Networking*, WON '08, pages 1678–1683, Okinawa, Japan, March 2008. IEEE Computer Society.

[42] Cyrus P. Hall, Antonio Carzaniga, Jeff Rose, and Alexander L. Wolf. A Content-Based Networking Protocol For Sensor Networks. Technical Report 0, Department of Computer Science, University of Colorado, August 2004.

[43] Sumi Helal. Standards for Service Discovery and Delivery. *IEEE Pervasive Computing*, 1(3):95–100, July 2002.

[44] Sumi Helal, Nitin Desai, Varun Verma, and Choonhwa Lee. Konark : Service Discovery and Delivery Protocol for Ad-hoc Networks. In *Proceedings of the 3rd IEEE Conference on Wireless Communication Networks*, WCNC '03, pages 2107–2113, New Orleans, USA, March 2003. IEEE.

[45] Pan Hui, J. Crowcroft, and E. Yoneki. BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks. *IEEE Transactions on Mobile Computing*, 10(11):1576–1589, November 2011.

[46] OPNET Technologies Inc. Opnet Modeler, 2004. http://www.opnet.com/products/modeler/home.html.

[47] Teerawat Issariyakul and Ekram Hossain. *Introduction to Network Simulator NS2*. Springer Publishing Company, Incorporated, 1st edition, 2010. http://www.isi.edu/nsnam/ns/.

[48] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a Delay Tolerant Network. In *Proceedings of the Annual Conference of the Special Interest Group on Data Communication*, SIGCOMM '04, pages 145–158, Portland,Oregon, USA, September 2004. ACM.

[49] Sushant Jain, Rahul C. Shah, Waylon Brunette, Gaetano Borriello, and Sumit Roy. Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks. *Mobile Networks and Applications*, 11(3):327–339, June 2006.

[50] Andrew Jenkins, SSebastian Kuzminsky, Kevin K. Gifford, Robert L. Pitts, and Kelvin Nichols. Delay/Disruption-Tolerant Networking: Flight Test Results from the International Space Station. In *Proceedings of the International Conference for Aerospace Experts, Academics, Military Personnel, and Industry Leaders*.

[51] Whitbeck. John, Yoann Lopez, Jeremie Leguay, Vania Conan, and Marcelo Dias de Amorim. Push-and-track: Saving Infrastructure Bandwidth Through Opportunistic Forwarding. *Pervasive and Mobile Computing*, 8(5):682–697, 2012.

[52] David Johnson and David MALTZ. Protocols for Adaptive Wireless and Mobile Networking. *IEEE Personal Communications*, 3:34–42, 1996.

[53] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. *ACM SIGARCH Computer Architure News*, 30(5):96–107, October 2002.

[54] Julien Haillot and Frédéric Guidec. A Protocol for Content-Based Communication in Disconnected Mobile Ad Hoc Networks. *Journal of Mobile Information Systems*, 6(2):123–154, 2010.

[55] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, Simutools '09. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[56] Michael Klein and Birgitta König-Ries. Multi-Layer Clusters in Ad-hoc Networks - An Approach to Service Discovery. In *Proceedings of the 1st International Workshop on Peer-to-Peer Computing*, IPTPS '02, pages 187–201, Cambridge, Massachusetts, USA, March 2002. Springer.

[57] Michael Klein, Birgitta König-Ries, and Philipp Obreiter. Service Rings - A Semantic Overlay for Service Discovery in Ad hoc Networks. In *Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, DEXA '03, pages 180–185, Washington, DC, USA, 2003. IEEE Computer Society.

[58] Matthias Klusch, Benedikt Fries, and Katia Sycara. Automated Semantic Web Service Discovery with OWLS-MX. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 915–922, New York, NY, USA, 2006. ACM.

[59] Young-Bae Ko and Nitin H. Vaidya. Location-aided Routing (LAR) in Mobile Ad Hoc Networks. *Wireless Networks*, 6(4):307–321, July 2000.

[60] UlaÅ§ C. Kozat and Leandros Tassiulas. Service Discovery in Mobile Ad hoc Networks: An Overall Perspective on Architectural Choices and Network Layer Support Issues. *Ad Hoc Networks*, 2(1):23–44, January 2004.

[61] Ulas Kozat and Leandros Tassiulas. Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *Proceedings of the 22nd International Conference on Computer Communications*, INFOCOM '03, pages 1965–1975. IEEE Societies, March 2003.

[62] Nicolas Le Sommer. A Framework for Service Provision in Intermittently Connected Mobile Ad hoc Networks. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, WOWMOM '07, pages 1–4, Helsinki, Finland, June 2007.

[63] Nicolas Le Sommer, Salma Ben Sassi, Frédéric Guidec, and Yves Mahéo. A Middleware Support for Location-Based Service Discovery and Invocation in Disconnected MANETs. *Studia Informatica Universalis*, 8(3):71–97, September 2010.

[64] Nicolas Le Sommer and Yves Mahéo. OLFServ: an Opportunistic and Location-Aware Forwarding Protocol for Service Delivery in Disconnected MANETs. In *Proceedings of the 5th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, UbiComm '11, pages 115–122, Lisbon, Portugal, November 2011. Xpert Publishing Services.

[65] Nicolas Le Sommer and Yves Mahéo. Location-Aware Routing for Service-Oriented Opportunistic Computing. *International Journal on Advances in Networks and Services*, 5(3):225–235, December 2012.

[66] Nicolas Le Sommer, Romeo Said, and Yves Mahéo. A Proxy-based Model for Service Provision in Opportunistic Networks. In *Proceedings of the 6th International Workshop on Middleware for Pervasive and Ad-Hoc Computing*, MPAC '08, Louvain, Belgique, December 2008. ACM Press.

[67] Choonhwa Lee and Sumi Helal. Context attributes: An Approach to Enable Context-awareness for Service Discovery. In *Proceedings of the International Conference on Applications and the Internet*, pages 22–30. IEEE, January 2003.

[68] Choonhwa Lee, Sumi Helal, and Wonjun Lee. Gossip-Based Service Discovery in Mobile Ad Hoc Networks. *The Institute of Electronics, Information and Communication Engineers (IEICE)*, pages 2621–2624, September 2006.

[69] Jérémie Leguay, Timur Friedman, and Vania Conan. DTN Routing in a Mobility Pattern Space. In *Proceedings of the the Annual Conference of the Special Interest Group on Data Communication, Workshop on Delay-tolerant Networking*, WDTN '05, pages 276–283, Philadelphia, Pennsylvania, USA, 2005. ACM.

[70] Vincent Lenders, Gunnar Karlsson, and Martin May. Wireless Ad Hoc Podcasting. In *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, SECON '07, pages 273–283, San Diego, California, USA, June 2007. IEEE Communications Society.

[71] Ilias Leontiadis, Paolo Costa, and Cecilia Mascolo. Extending Access Point Connectivity Through Opportunistic Routing in Vehicular Networks. In *Proceedings of the 29th Conference on Computer Communications*, INFOCOM '10, pages 486–490, San Diego, California, USA, 2010. IEEE Press.

[72] Qinghua Li, Sencun Zhu, and Guohong Cao. Routing in Socially Selfish Delay Tolerant Networks. In *Proceedings of the 29th Conference on Computer Communications*, INFOCOM '10, pages 857–865, San Diego, California, USA, 2010. IEEE Press.

[73] Yong Li, Mengjiong Qian, Depeng Jin, Pan Hui, Zhaocheng Wang, and Sheng Chen. Multiple Mobile Data Offloading Through Disruption Tolerant Networks. *IEEE Transactions on Mobile Computing*, 13(7):1–1, 2014.

[74] Wen-Hwa Liao, Jang-Ping Sheu, and Yu-Chee Tseng. GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks. *Telecommunication Systems*, 18(1-3):37–60, 2001.

[75] Wen-Hwa Liao, Yu-Chee Tseng, Kuo-Lun Lo, and Jang-Ping Sheu. GeoGRID: A Geocasting Protocol For Mobile Ad Hoc Networks Based on GRID. *Internet Technologies*, 1(2):23–32, 2000.

[76] Anders Lindgren, Avri Doria, and Samo Grasic. Probabilistic Routing Protocol for Intermittently Connected Networks. RFC 6693, IRTF, February 2012.

[77] Anders Lindgren, Avri Doria, and Olov Schelen. Probabilistic Routing in Intermittently Connected Networks. In *Service Assurance with Partial and Intermittent Resources*, volume 3126 of *Lecture Notes in Computer Science*, pages 239–254. Springer Berlin Heidelberg, 2004.

[78] Ting Liu, Christopher M. Sadler, Pei Zhang, and Margaret Martonosi. Implementing Software on Resource-constrained Mobile Sensors: Experiences with Impala and ZebraNet. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, MobiSys '04, pages 256–269, Boston, MA, USA, 2004. ACM.

[79] Honghui Luo and Michel Barbeau. Performance Evaluation of Service Discovery Strategies in Ad Hoc Networks. *Communication Networks and Services Research, Annual Conference on*, pages 61–68, 2004.

[80] Mathew MacKenzie, Ken Laskey, Francis McCabe, Peter Brown, and Rebekah Metz. Reference Model for Service Oriented Architecture 1.0. Technical report, OASIS, February 2006.

[81] Yves Mahéo and Romeo Said. Service Invocation over Content-Based Communication in Disconnected Mobile Ad Hoc Networks. In *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications*, AINA '10, pages 503–510, Washington, DC, USA, 2010. IEEE Computer Society.

[82] Mahesh K. Marina and Samir R. Das. On-Demand Multi Path Distance Vector Routing in Ad Hoc Networks. In *Proceedings of the 9th International Conference on Network Protocols*, ICNP '01, Washington, DC, USA, 2001. IEEE Computer Society.

[83] Ramon Martí, Sergi Robles, Abraham Martín-Campillo, and Jordi Cucurull. Providing early resource allocation during emergencies: The mobile triage tag. *Journal of Network and Computer Applications*, 32(6):1167–1182, 2009.

[84] Abraham Martín-Campillo and Ramon Martí. Energy-efficient forwarding mechanism for wireless opportunistic networks in emergency scenarios. *Computer Communications*, 35(14):1715–1724, 2012. Special issue: Wireless Green Communications and Networking.

[85] Rubén Martinez-Vidal, Ramon Martí, and Joan Borrell. Characterization of a Transoceanic Aircraft Delay Tolerant Network. In *Proceedings of the 38th Conference on Local Computer Networks*, LCN '13, pages 565–572, October 2013.

[86] Rubén Martinez-Vidal, Ramon Martí, and Joan Borrell. Analyzing Information Propagation in a Transoceanic Aircraft Delay Tolerant Network. In *Proceedings of the 39th Conference on Local Computer Networks*, LCN '14, pages 116–123, September 2014.

[87] Sun Microsystems. JINI Architecture Specification. November 1999.

[88] Shinji Motegi, Kiyohito Yoshihara, and Hiroki Horiuchi. Service Discovery for Wireless Ad hoc Networks. In *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications*, pages 232–236, Honolulu, HI, USA, October 2002. IEEE.

[89] Mirco Musolesi and Cecilia Mascolo. CAR: Context-Aware Adaptive Routing for Delay Tolerant Mobile Networks. *IEEE Transactions on Mobile Computing*, 8(2):246–260, 2009.

[90] National Institute of Standards and Technology (NIST). SHA-1 Standard, sep 2001.

[91] Andronikos Nedos, Kulpreet Singh, and Siobhán Clarke. Service*: Distributed Service Advertisement for Multi-Service, Multi-Hop MANET Environments. In *Proceedings of 7th IFIP International Conference on Mobile and Wireless Communication Networks*, MWCN '05.

[92] Hoang Anh Nguyen, Silvia Giordano, and Alessandro Puiatti. Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Network (PROPICMAN). In *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks*, WoWMoM '07, pages 1–6, Helsinky, Finland, June 2007. IEEE Computer Society.

[93] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '99, pages 151–162. ACM/IEEE, August 1999.

[94] Michael Nidd. Service discovery in DEAPspace. *Personal Communications, IEEE*, 8(4):39–45, 2001.

[95] Hervé Ntareme and Sebastian Domancich. Security and Performance Aspects of Bytewalla: A Delay Tolerant Network on Smartphones. In *Proceedings of the 7th International Conference on Wireless and Mobile Computing, Networking and Communications*, WiMob '11, pages 449–454, Shanghai, China, October 2011. IEEE.

[96] Hervé Ntareme, Marco Zennaro, and Björn Pehrson. Delay Tolerant Network on Smartphones: Applications for Communication Challenged Areas. In *Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition*, ExtremeCom '11, pages 14:1–14:6, New York, NY, USA, 2011. ACM.

[97] Jörg Ott and Dirk Kutscher. Bundling the Web: HTTP over DTN. In *Proceedings of the Workshop on Networking in Public Transport*, WNEPT '06, Waterloo, Ontario, Canada, August 2006.

[98] Mike P. Papazoglou. Service-Oriented Computing: Concepts, Characteristics and Directions. pages 3–12, December 2003.

[99] Eugster Patrick, Pascal Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The Many Faces of Publish/Subscribe. *ACM Computing Surveys*, 35:114–131, 2003.

[100] Alex (Sandy) Pentland, Richard Fletcher, and Amir Hasson. DakNet: Rethinking Connectivity in Developing Nations. *Computer*, 37(1):78–83, January 2004.

[101] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computer Systems and Applications*, WMCSA '99, New Orleans, Louisiana, USA, February 1999. IEEE Computer Society.

[102] Mikko Pitkänen, Teemu Kärkkäinen, Jöerg Ott, Marco Conti, Andrea Passarella, Silvia Giordano, Daniele Puccinelli, Franck Legendre, Sasha Trifunovic, Karin Anna Hummel, Martin May, Nidhi Hegde, and Thrasyvoulos Spyropoulos. SCAMPI: Service Platform for Social Aware Mobile and Pervasive Computing. In *Proceedings of the 1st Mobile Cloud Computing Workshop*, MCC '12, pages 7–12, Helsinki, Finland, August 2012. ACM.

[103] Mikko Pitkänen, Teemu Kärkkäinen, and Jörg Ott. Mobility and Service Discovery in Opportunistic Networks. *IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 204–210, 2012.

[104] Jonathan Postel. User Datagram Protocol, August 1980.

[105] Olga Ratsimor, Dipanjan Chakraborty, Anupam Joshi, and Timothy Finin. Allia: Alliance-based Service Discovery for Ad-hoc Environments. In *Proceedings of the 2nd International Workshop on Mobile Commerce*, WMC '02, pages 1–9, Atlanta, Georgia, USA, September 2002. ACM.

[106] Nishkam Ravi, Peter Stern, Niket Desai, and Liviu Iftode. Accessing Ubiquitous Services Using Smart Phones. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, Kauai, Hawaii, 2005.

[107] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seongjoon Kim, and Song Chong. CRAWDAD data set ncsu/mobility models (v. 2009-07-23). Downloaded from http://crawdad.cs.dartmouth.edu/ncsu/mobilitymodels, July 2009.

[108] Obay Sabrie, Hasan Hasan, and Rosli Salleh. Fast Handoff for 802.11 Wireless Network. *Communications and Network*, 3(4), October 2011.

[109] Francoise Sailhan and Valerie Issarny. Scalable Service Discovery for MANET. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, PERCOM '05, pages 235–244, Kauai, Hawaii, 2005. IEEE Computer Society.

[110] Jordy Sangers, Flavius Frasincar, Frederik Hogenboom, and Vadim Chepegin. Semantic Web Service Discovery Using Natural Language Processing Techniques. *Expert Systems with Applications*, 40(11):4660–4671, 2013.

[111] Gregor Schiele, Christian Becker, and Kurt Rothermel. Energy-Efficient Cluster-based Service Discovery for Ubiquitous Computing. In *Proceedings of Annual Conference of the Special Interest Group on Data Communication, 11th Workshop on ACM SIGOPS European Workshop*, EW '04, page 14, Leuven, Belgium, September 2004. ACM.

[112] Keith Scott and Scott Burleigh. Bundle Protocol Specification. RFC 5050 (Experimental), November 2007.

[113] Karim Seada and Ahmed Helmy. Rendezvous Regions: A Scalable Architecture for Service Location and Data-Centric Storage in Large-Scale Wireless Networks. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium*, IPDPS '04, Santa Fe, New Mexico, April 2004. IEEE Computer Society.

[114] Amin Seyed Seno, Hosseini, Rahmat Budiarto, and Tat-Chee Wan. Survey and new Approach in Service Discovery and Advertisement for Mobile Ad hoc Networks. *International Journal of Computer Science and Network Security*, 7(2):275–284, 2007.

[115] Siva Sivavakeesar, Oscar F. Gonzalez, and George Pavlou. Service Discovery Strategies in Ubiquitous Communication Environments. *IEEE Communications Magazine*, 44(9):106–113, September 2006.

[116] Tara Small and Zygmunt J. Haas. The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, MobiHoc '03, pages 233–244, Annapolis, Maryland, USA, June 2003. ACM.

[117] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi Raghavendra. Single-Copy Routing in Intermittently Connected Mobile Networks. In *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, SECON '04, pages 235–244, Santa Clara, California, October 2004. IEEE.

[118] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *Proceedings of the the Annual Conference of the Special Interest Group on Data Communication, Workshop on Delay-tolerant Networking*, WDTN '05, pages 252–259. ACM, August 2005.

[119] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility. In *Proceedings of the 5th IEEE International Conference on Pervasive Computing and Communications Workshops*, PERCOMW '07, pages 79–85, White Plains, NY, USA, March 2007. IEEE Computer Society.

[120] W. Richard Stevens. *TCP/IP Illustrated (Vol. 1): The Protocols*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1993.

[121] Sacha Trifunovic, Bernhard Distl, Dominik Schatzmann, and Franck Legendre. WiFi-Opp: Ad-hoc-less Opportunistic Networking. In *Proceedings of the 6th ACM Workshop on Challenged Networks*, CHANTS '11, pages 37–42, Las Vegas, Nevada, USA, 2011. ACM.

[122] Jerry Tyan and Qusay H. Mahmoud. A Comprehensive Service Discovery Solution for Mobile Ad Hoc Networks. *Mobile Networks and Applications*, 10(4), August 2005.

[123] Amin Vahdat and David Becker. Epidemic Routing for Partially-Connected Ad Hoc Networks. Technical report, July 2000.

[124] Andras Varga. The omnet++ discrete event simulation system. *Proceedings of the European Simulation Multiconference*, June 2001.

[125] Christopher Ververidis and George Polyzos. Routing Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications Workshops*, PerCom '05, pages 258–262, Kauai, Hawaii, March 2005.

[126] Christopher N. Ververidis and George C. Polyzos. Service Discovery for Mobile Ad Hoc Networks: A Survey of Issues and Techniques. *IEEE Communications Surveys and Tutorials*, 10(3):30–45, 2008.

[127] Long Vu, Quang Do, and Klara Nahrstedt. 3R: Fine-grained encounter-based routing in Delay Tolerant Networks. In *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks*, WoWMoM '11, pages 1–6, June 2011.

[128] W3C CONSORTIUM. OWL Web Ontology Language. W3C Recommendation, February 2004. http://www.w3.org/TR/owl-features/.

[129] W3C CONSORTIUM. Semantic Annotations for WSDL and XML Schema. W3C Recommendation, August 2007. http://www.w3.org/TR/sawsdl/.

[130] Mark Weiser. The Computer of the Twenty-First Century. *Scientific American*, 265(3):66–75, September 1991.

[131] Mark Weiser. Hot Topics: Ubiquitous Computing. *IEEE Computer Communications Magazine*, 26(10):71–72, October 1993.

[132] Mark Weiser. Some Computer Science Issues in Ubiquitous Computing. *Commununications of the ACM*, 36(7):75–84, July 1993.

[133] Lloyd Wood, Will Ivancic, Wesley M. Eddy, Dave Stewart, James Northam, Chris Jackson, and Alex Da Silva Curiel. Use of The Delay-Tolerant Networking Bundle Protocol From Space. Proceedings of the 59th International Astronautical Congress, September 2008.

[134] Jie Wu and Fei Dai. Broadcasting in Ad Hoc Networks Based on Self-Pruning. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM '03, pages 2240–2250, San Francisco, California, USA, March 2003. IEEE Computer Society.

[135] Eiko Yoneki, Pan Hui, ShuYan Chan, and Jon Crowcroft. A Socio-aware Overlay for Publish/Subscribe Communication in Delay Tolerant Networks. In *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, MSWiM '07, pages 225–234, New York, NY, USA, 2007. ACM.

[136] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks. *ACM SIGSIM Simulation Digest*, 28(1), July 1998.

[137] Ying Zhang, Hui He, and Jing Teng. Chord-Based Semantic Service Discovery with QoS. In *Proceedings of the 5th International Conference on Measuring Technology and Mechatronics Automation*, ICMTMA '13, pages 365–367, Hong Kong, China, January 2013.

[138] Wenrui Zhao, Mostafa Ammar, and Ellen Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '04, pages 187–198, Roppongi Hills, Tokyo, Japan, May 2004.

[139] Feng Zhu, Matt Mutka, and Lionel Ni. PrudentExposure: A Private and User-centric Service Discovery Protocol. In *Proceedings of the 2nd IEEE International Conference on Pervasive Computing and Communications*, pages 329–338, March 2004.

Publications

[1] Ali Makke, Nicolas Le Sommer, and Yves Mahéo. TAO: A Time-Aware Opportunistic Routing Protocol for Service Invocation in Intermittently Connected Networks. In *8th International Conference on Wireless and Mobile Communications (ICWMC 2012)*, pages 118–123, Venise, Italy, June 2012.

[2] Nicolas Le Sommer, Ali Makke, and Yves Mahéo. A Soft Handover for Service Delivery in Intermittently Connected Hybrid Networks. In *5th International Conference on Mobile Wireless Middleware, Operating Systems, and Applications (Mobilware 2012)*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pages 122–135, Berlin, Germany, November 2012. Springer Berlin Heidelberg.

[3] Ali Makke, Yves Mahéo, and Nicolas Le Sommer. Towards Opportunistic Service Provisioning in Intermittently Connected Hybrid Networks. In *4th International Conference on Networking and Distributed Computing (ICNDC 2013)*, pages 28–32, Honk Kong, China, December 2013. IEEE CS.

Résumé

La vision de l'informatique ubiquitaire permettant de construire des espaces intelligents interactifs dans l'environnement physique passe, peu à peu, du domaine de la recherche à la réalité. La capacité de calcul ne se limite plus à l'ordinateur personnel mais s'intègre dans de multiples appareils du quotidien, et ces appareils deviennent, grâce à plusieurs interfaces, capables de communiquer directement les uns avec les autres ou bien de se connecter à Internet.
Dans cette thèse, nous nous sommes intéressés à un type d'environnement cible de l'informatique ubiquitaire qui forme ce que nous appelons un réseau hybride à connexions intermittentes (ICHN). Un ICHN est un réseau composé de deux parties : une partie fixe et une partie mobile. La partie fixe est constituée de plusieurs infostations fixes (potentiellement reliées entre elles avec une infrastructure fixe, typiquement l'Internet). La partie mobile, quant à elle, est constituée de smartphones portés par des personnes nomades. Tandis que la partie fixe est principalement stable, la partie mobile pose un certain nombre de défis propres aux réseaux opportunistes. En effet, l'utilisation de moyens de communication à courte portée couplée à des déplacements de personnes non contraints et à des interférences radio induit des déconnexions fréquentes. Le concept du "store, carry and forward" est alors habituellement appliqué pour permettre la communication sur l'ensemble du réseau. Avec cette approche, un message peut être stocké temporairement sur un appareil avant d'être transféré plus tard quand les circonstances sont plus favorables. Ainsi, n'importe quel appareil devient un relai de transmission opportuniste qui permet de faciliter la propagation d'un message dans le réseau.
Dans ce contexte, la fourniture de services est particulièrement problématique, et exige de revisiter les composants principaux du processus de fourniture, tels que la découverte et l'invocation de service, en présence de ruptures de connectivité et en l'absence de chemins de bout en bout. Cette thèse aborde les problèmes de fourniture de service sur l'ensemble d'un ICHN et propose des solutions pour la découverte de services, l'invocation et la continuité d'accès.
En ce qui concerne le défi de la découverte de services, nous proposons TAO-DIS, un protocole qui met en ¡uvre un mécanisme automatique et rapide de découverte de services. TAO-DIS tient compte de la nature hybride d'un ICHN et du fait que la majorité des services sont fournis par des infostations. Il permet aux utilisateurs mobiles de découvrir tous les services dans l'environnement afin d'identifier et de choisir les plus intéressants.
Pour permettre aux utilisateurs d'interagir avec les services découverts, nous introduisons TAO-INV. TAO-INV est un protocole d'invocation de service spécialement conçu pour les ICHN. Il se fonde sur un ensemble d'heuristiques et de mécanismes qui assurent un acheminement efficace des messages (des requêtes et des réponses de services) entre les infostations fixes et les clients mobiles tout en conservant un surcoût et des temps de réponses réduits.
Puisque certaines infostations dans le réseau peuvent être reliées entre elles, nous proposons un mécanisme de continuité d'accès (handover) qui modifie le processus d'invocation pour réduire les délais de délivrance. Dans sa définition, il est tenu compte de la nature opportuniste de la partie mobile de l'ICHN.
Nous avons mené diverses expérimentations pour évaluer nos solutions et les comparer à d'autres protocoles conçus pour des réseaux ad hoc et des réseaux opportunistes. Les résultats obtenus tendent à montrer que nos solutions surpassent ces autres protocoles, notamment grâce aux optimisations que nous avons développées pour les ICHN. À notre avis, construire des protocoles spécialisés qui tirent parti des techniques spécifiquement conçues pour les ICHN est une approche à poursuivre en complément des recherches sur des protocoles de communication polyvalents.

Abstract

The vision of pervasive computing of building interactive smart spaces in the physical environment is gradually heading from the research domain to reality. Computing capacity is moving beyond personal computers to many day-to-day devices, and these devices become, thanks to multiple interfaces, capable of communicating directly with one another or of connecting to the Internet.
In this thesis, we are interested in a kind of pervasive computing environment that forms what we call an Intermittently Connected Hybrid Network (ICHN). An ICHN is a network composed of two parts: a fixed and a mobile part. The fixed part is formed of some fixed infostations (potentially connected together with some fixed infrastructure, typically the Internet). The mobile part, on the other hand, is formed of smartphones carried by nomadic people. While the fixed part is mainly stable, the mobile part is considered challenging and form what is called an Opportunistic Network. Indeed, relying on short-range communication means coupled with the free movements of people and radio interferences lead to frequent disconnections. To perform a network-wide communication, the "store, carry and forward" approach is usually applied. With this approach, a message can be stored temporarily on a device, in order to be forwarded later when circumstances permit. Any device can opportunistically be used as an intermediate relay to facilitate the propagation of a message from one part of the network to another.
In this context, the provisioning of pervasive services is particularly challenging, and requires revisiting important components of the provisioning process, such as performing pervasive service discovery and invocation with the presence of connectivity disruptions and absence of both end-to-end paths and access continuity due to user mobility. This thesis addresses the problems of providing network-wide service provisioning in ICHNs and proposes solutions for pervasive service discovery, invocation and access continuity.
Concerning service discovery challenge, we propose TAO-DIS, a service discovery protocol that performs an automatic and fast service discovery mechanism. TAO-DIS takes into account the hybrid nature of an ICHN and that the majority of services are provided by infostations. It permits mobile users to discover all the services in the surrounding environment in order to identify and choose the most convenient ones.
To allow users to interact with the discovered services, we introduce TAO-INV. TAO-INV is a service invocation protocol specifically designed for ICHNs. It relies on a set of heuristics and mechanisms that ensures performing efficient routing of messages (both service requests and responses) between fixed infostations and mobile clients while preserving both low values of overhead and round trip delays.
Since some infostations in the network might be connected, we propose a soft handover mechanism that modifies the invocation process in order to reduce service delivery delays. This handover mechanism takes into consideration the opportunistic nature of the mobile part of the ICHN.
We have performed various experiments to evaluate our solutions and compare them with other protocols designed for ad hoc and opportunistic networks. The obtained results tend to prove that our solutions outperform these protocols, namely thanks to the optimizations we have developed for ICHNs. In our opinion, building specialized protocols that benefit from techniques specifically designed for ICHNs is an approach that should be pursued, in complement with research works on general-purpose communication protocols.

UBS Université de Bretagne-Sud

n d'ordre : 223

Université de Bretagne Sud

Centre d'Enseignement et de Recherche Y. Coppens - rue Yves Mainguy - 56000 VANNES
Tél : + 33(0)2 97 01 70 70 Fax : + 33(0)2 97 01 70 70