From traditional to interesting: How Interests change the Routing decisions and Performance metrics in Opportunistic Networks

María Daniela Córdova Pintado

UAB
Universitat Autònoma
de Barcelona

Universitat Autònoma de Barcelona

Departament d'Enginyeria de la Informació i de les Comunicacions

PHD in Computer Science

From traditional to interesting: How Interests change the Routing decisions and Performance metrics in Opportunistic Networks

María Daniela Córdova Pintado

Supervisors
Dr. Adrián Sánchez Carmona
Dr. Ramon Martí Escalé

Bellaterra, February 2024

This thesis was typeset with LaTeX 2$_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.

Front Cover Image:

`https://wordclouds.ethz.ch/`

I certify that I have read this thesis entitled “ From traditional to interesting: How Interests change the Routing decisions and Performance metrics in Opportunistic Networks" and that, in my opinion, it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Bellaterra, February 2024

Dr. Adrián Sánchez Carmona
(Advisor)

I certify that I have read this thesis entitled “ From traditional to interesting: How Interests change the Routing decisions and Performance metrics in Opportunistic Networks" and that, in my opinion, it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Bellaterra, February 2024

Dr. Ramon Martí Escalé
(Advisor)

Committee:
- Jaime Delgado Mercé
- Guillermo Navarro Arribas
- Carles Garrigues Olivella

Substitute:
- Ruben Tous Liesa
- David Castells Rufas

To Jesus

Abstract

THis thesis presents all the work conducted in order to design and develop a new strategy for operating in Opportunistic Networks (*OppNets*), which allows efficient management of node resources by using the users' interest to route messages throughout the network.

We present Interest-based Routing Protocol (*IRP*), that uses the users' interests in a topic to identify the destination of each message, adapting to real-time changes that may occur. Additionally, since *IRP* knows the interests of its neighbors in each opportunistic contact, the nodes that use this strategy build a perception of interest for each topic in the network.

IRP uses this information to make forwarding decisions about messages only when and where there is interest in a topic, and to remove the less interesting messages from the network.

In order to measure the performance in terms of message Delivery Ratio and Latency, we redefine these two metrics to support the user's changing interests. In both cases, they only take into account the periods where users are interested in receiving messages.

Finally, we conduct several experiments to evaluate *IRP*, including two synthetic scenarios that use the real map of Barcelona and one scenario with the real contact traces of an academic course at MIT. In these scenarios, we run a set of simulations and compare *IRP* with an interest-based algorithm (Social-aware Content-based Opportunistic Routing Protocol (*SCORP*)) and the two opportunistic algorithms traditionally used (Epidemic (*EP*) and Spray & Wait (*S&W*)). In the experiments, *IRP* achieve better results in terms of Messages Delivered, Messages Relayed, Delivery Ratio, Network Overhead and Buffer Occupancy, while also showing the best Efficiency throughout the simulations.

Resum

AQUESTA tesi presenta tot el treball realitzat per dissenyar i desenvolupar una nova estratègia per operar en Xarxes Oportunistes *(OppNets)*, que permet una gestió eficient dels recursos dels nodes utilitzant l'interès dels usuaris per encaminar els missatges a través de la xarxa.

Presentem Protocol d'Enrutament basat en Interessos *IRP*, que utilitza els interessos dels usuaris en un tema per identificar la destinació de cada missatge, adaptant-se als canvis en temps real que puguin produir-se. A més, atès que IRP coneix els interessos dels seus veïns en cada contacte oportunista, els nodes que utilitzen aquesta estratègia construeixen una percepció d' interès per a cada tema a la xarxa.

IRP utilitza aquesta informació per prendre decisions de reenviament de missatges només quan i on hi ha interès en un tema, i per eliminar els missatges menys interessants de la xarxa.

Per mesurar el rendiment en termes de Ràtio de Lliurament de missatges i Latència, redefinim aquestes dues mètriques perquè tinguin en compte els interessos canviants de l'usuari. En tots dos casos, només tenen en compte els períodes en què els usuaris estan interessats a rebre missatges sobre un tema concret.

Finalment, realitzem diversos experiments per avaluar *IRP*, incloent-hi dos escenaris sintètics que utilitzen el mapa real de Barcelona i un escenari amb les traces de contactes reals d'un curs acadèmic al MIT. En aquests escenaris, executem un conjunt de simulacions i comparem *IRP* amb un algorisme basat en interessos (Social-aware Content-based Opportunistic Routing Protocol (*SCORP*)) i els dos algoritmes oportunistes utilitzats tradicionalment (Epidemic (*EP*) i Spray & Wait (*S&W*)). En els experiments, *IRP* aconsegueix millors resultats en termes de Missatges Lliurats, Missatges Retransmesos, Ràtio de Lliurament, Sobrecàrrega de Xarxa i Ocupació de Buffer, mostrant a més la millor Eficiència al llarg de les simulacions.

Resumen

ESTA tesis presenta todo el trabajo realizado para diseñar y desarrollar una nueva estrategia para operar en Redes Oportunistas *(OppNets)*, que permite una gestión eficiente de los recursos de los nodos utilizando el interés de los usuarios para encaminar los mensajes a través de la red.

Presentamos Protocolo de Enrutamiento basado en Intereses *(IRP)*, que utiliza los intereses de los usuarios en un tema para identificar el destino de cada mensaje, adaptándose a los cambios en tiempo real que puedan producirse. Además, dado que *IRP* conoce los intereses de sus vecinos en cada contacto oportunista, los nodos que utilizan esta estrategia construyen una percepción de interés para cada tema en la red.

IRP utiliza esta información para tomar decisiones de reenvío de mensajes sólo cuando y donde hay interés en un tema, y para eliminar los mensajes menos interesantes de la red.

Para medir el rendimiento en términos de Ratio de Entrega de mensajes y Latencia, redefinimos estas dos métricas para que tengan en cuenta los intereses cambiantes del usuario. En ambos casos, sólo tienen en cuenta los periodos en los que los usuarios están interesados en recibir mensajes sobre un tema concreto.

Por último, realizamos varios experimentos para evaluar *IRP*, incluyendo dos escenarios sintéticos que utilizan el mapa real de Barcelona y un escenario con las trazas de contactos reales de un curso académico en el MIT. En estos escenarios, ejecutamos un conjunto de simulaciones y comparamos *IRP* con un algoritmo basado en intereses (Social-aware Content-based Opportunistic Routing Protocol (*SCORP*)) y los dos algoritmos oportunistas utilizados tradicionalmente (Epidemic (*EP*) y Spray & Wait (*S&W*)). En los experimentos, *IRP* logra mejores resultados en términos de Mensajes Entregados, Mensajes Retransmitidos, Ratio de Entrega, Sobrecarga de Red y Ocupación de Buffer, mostrando además la mejor Eficiencia a lo largo de las simulaciones.

Acknowledgements

THANKS to everyone who has been part of this unforgettable adventure of continuing to learn. I am infinitely grateful to God for having put it in my heart to turn a dream into reality.

If I had to write the story during this journey, I would say that I am just another character in the cast, I would say that the protagonists of this story are all those who left memories of them in me. Ramon and Adrián are the main ones, with their strict dedication, knowledge, and patience they transformed me and made of white pages a work to be read, not only for the work, but also for how they have forged my character, just as my dear Ing. Hallo did one day when I was still a young girl.

To you, Sergi, thank you for saying yes. A yes that changed the life of four people, some years ago. Thank you for listening to me when everything was difficult. Thank you for your trust.

In this story, my gratitude to my husband Milton, who silently listened and was part of all my ups and downs. His story is surely a different version of mine. Thank you, my love.

Thanks to the two who cannot be missing in this story, my daughters Isabel and Zara, who left love notes in my work notes. When I read them, they encouraged me to keep going. This is for you. Thank you, my babies.

My daddies, my daddies, thanks to them. Daddy, you with your smile and your encouragement, your help in everything as if I were still your little girl at school. Mamita linda, I saw you every day, I listened to you every day, without your prayers, your wisdom and your encouragement this would not have been possible. Although you are far away, you are my strength, half daddy, half mommy.

To my family in Catalonia, thank you Nuria, Laura and Daniel for being part of our life and also of this story.

Finally, thanks to my partners, mom Lili, Edy, Javi, JuanCa, Patito and especially to my beloved Poli.

Gracias Totales.

Daniela
Bellaterra, February 2024

Contents

III Experimentation and Results: *IRP*'s performance 37

List of Figures

List of Tables

List of Algorithms

Abbreviations list

IRP	Interest-based Routing Protocol
$S\&W$	Spray And Wait Routing Protocol
EP	Epidemic Routing Protocol
$SCORP$	Social-aware Content-based Opportunistic Routing Protocol
t	A topic
A	Node A
B	Node B
I_t^N	Interest in a topic
P_t^N	Perception of Interest in a topic
V_t^B	Views on a topic
β	Alpha parameter
α	Beta parameter
T_f	Forwarding threshold
m	A message
DR	Delivery Ratio
DR'	Delivery Ratio using users interest
m_d	Delivered messages on every topic
m_{pd}	Potential number of messages that could be delivered
m_d^t	Additive increase factor
m_{pd}^t	Potential messages that could be delivered in every topic
$m_{pd}^t(int)$	Potential number of messages that could be delivered to nodes currently interested in a topic
$m_{pd}^t(\neg int)$	Potential number of messages that could have been delivered to nodes that were interested at some point, but they are no longer
m_g^t	Number of messages generated on a topic
$\#int$	Number of nodes currently interested in that topic
$m_{pd}^t(\neg int)$	Number of messages generated that could have been delivered to the nodes that are no longer interested in a topic
$m_g^t(time_{N\neg})$	Number of messages that were generated in topic t until the moment when node N became disinterested in it
L_m	Latency of a message
T_{m_g}	Time elapsed between the moment when a message was generated
T_{m_d}	The moment when a message is delivered
L'_m	Latency considering the users interest
$\bar{P_t}$	The set containing all periods of time
p	A period of time
T_{m_g}	Time in which a messages is generated
T_{m_d}	Time in which a messages is delivered
$T_{I_t^N}(init)_p$	Time in which the interests in a topic starts
$T_{I_t^N}(end)_p$	Time in which the interests in a topic ends
L	Number of copies in Spray & Wait ($S\&W$)
w	Width
h	Height

Part I

Preliminaries

Introduction

1

LIVING in an age where the evolution and rapid advancement of smart devices have become an integral part of our lives, people are becoming more and more dependent on their devices and the countless applications they offer to share thoughts, experiences, interests, among others.

Nowadays, 𝕏, previously known as twitter, is an example of these applications, where senders use a #hashtag to define a message topic, and receivers filter the messages they want to read according to the topics they are interested in. These types of applications operate in connected and centralized architectures, and can sometimes face problems like censorship, or monopolies that lead to discrimination or exclusion of users or messages for economic, political or other reasons by governments or network/application's operators. These problems reveal the need for the development of decentralized networks like the Opportunistic Networks (*OppNets*), since they minimize those risks and allow users to interchange information without need of a centralized architecture. Even in situations of total or partial destruction of the infrastructure by natural disasters [27], Opportunistic Networks (*OppNets*) offer an opportunity of interaction between users.

There are some messaging applications like Bridgefy[1] or the discontinued Firechat[2] that have been designed to operate when there is no fixed connectivity infrastructure through sending messages to neighboring devices. Other applications, like the Opportunistic Twitter [34], that works in hybrid networks, where users can also exchange tweets when they meet opportunistically. They all base their design on the *store, carry and forward OppNets* [1] paradigm.

In any case, all these applications use basic routing strategies that tend to flood the network with messages, regardless of the number of users that may be interested in them, and consume significant device and network resources. Therefore, they tend to be inefficient in message delivery in *OppNets*, which becomes especially evident in situations where users' interests change in real time and where network resources are limited. This is something worth improving, since *OppNets* users with low resources may become selfish and not cooperate with the network by forwarding messages for the sake of other users [36].

---

[1] https://github.com/bridgefy

[2] https://edition.cnn.com/2014/10/16/tech/mobile/tomorrow-transformed-firechat

Based on all those premises, when we analyze the existing solutions, we see that pure *OppNets* solutions that allow users to send messages on any topic to multiple destinations that can change at any time have not been presented yet. Also, there are no solutions in which users can change their interests at any time. Besides, most systems do not tie messages' lifespan with networks' interests. Therefore, it would be very helpful to provide the developers with tools that efficiently allow all of this.

In order to make this possible and solve some of the problems outlined above, we define the concepts of *Interest in a topic* and *Perception of Interest in a topic*, and use them to propose the Interest-based Routing Protocol (*IRP*).

Thus, the main contribution of this work is the design of the Interest-based Routing Protocol (*IRP*), a new strategy that uses the users' interest in a topic for making forwarding decisions. This new strategy allows users to change their *Interest in a topic* in real time, and therefore it adapts to the fact that message destinations can change at any moment. It aims to deliver as many messages as possible on topics with high global interest, improving the use of network resources when or where there is not enough, and preventing the network from being flooded with unnecessary messages. In addition, this strategy allows users to create topics dynamically, and removes the messages from the network when there is no more global interest in them.

When a new routing algorithm in *OppNets* is presented, it is a good practice to evaluate it in order to determine its efficiency against other proposals. In situations where users' interest changes in real time (nodes become interested, then disinterested, and then interested again, for example), metrics like Delivery Ratio and Message Latency may not provide good enough information to measure performance because they do not take into account the changing interests of users. So, along with our routing proposal, we also propose a new method to calculate the Delivery Ratio and the Messages Latency metrics, because in scenarios where the potential number of message destinations changes in real-time and there are moments where messages are not delivered because users are not interested, these metrics should not be calculated in the traditional way.

To evaluate the new routing protocol, we designed a set of experiments using The *ONE* [20] simulator, in which we compared *IRP* with Epidemic [42], Spray & Wait [39] and Social-aware Content-based Opportunistic Routing Protocol [29]. We found that *IRP* has good overall results on Messages Delivered, Messages Relayed, Delivery Ratio, Buffer Occupancy, and its Efficiency is the best throughout the experiments.

1.1 Objectives

The objective of our work is to improve the efficiency of message transmission in Opportunistic Networks (*OppNets*) by using the users' interest. To achieve our objective, we deploy it along two main lines:

- **The definition and development of a new Routing Strategy:** To use the users' interest in a topic in order to propose a new routing algorithm for improving the consumption of network resources in Opportunistic Networks (*OppNets*) and to deliver as many messages as possible when and where there are more people interested in the network.

- **The redefinition of traditional Performance Metrics:** To formulate the way of calculating traditional performance metrics such as Delivery Ratio and Latency to adapt them to situations where the destinations of a message change in real time according to the interest of the users.

1.2 Contributions

Based on the previous objectives, the contributions of this thesis are:

1. The Interest-based Routing Protocol (*IRP*), a new routing algorithm to improve the efficiency in relaying messages in Opportunistic Networks (*OppNets*) using the users' interest.

    - Allowing the real-time changes in users' interests on topics and to deliver as many interesting messages as possible.

    - Improving the use of network resources, relaying only when and where there are interests.

    - Allowing messages to remain on the network as long as there is interest in them.

2. A new way to calculate the traditional metrics in Opportunistic Networks (*OppNets*) by taking into account the users' interest.

    - Obtaining better information about the performance of the algorithms, taking into account only the moments when users are interested, regardless of how many times those users change their interest in real time.

1.3 Document Layout

The remainder of this document is organized as follows. In Part I, we present an Introduction of our research work and the Related Work about the existing routing and metrics solutions. Then, in Part II, we present our Contributions, the new Interest-based Routing Protocol and detail its design. Also, we present the two redefined Metrics; the first one is the Delivery Ratio and the second one the Latency. Then, in Part III, we present the different Experiments designed to evaluate the new proposal, and discuss the Results obtained with the help of the redefined metrics. In Part IV, we draw the Conclusions and mention some Future lines of work. And, finally, in Part V, we detail the Bibliography of this work.

1.4 List of Publications

As a result of the work of this thesis, we wrote two articles that have been published in international conferences:

[1] Daniela Córdova-Pintado, Adrián Sánchez-Carmona and Ramon Martí, **“Paving the way towards delivering messages based on user interests in *OppNets*"** in 2022, 18th International Conference on Wireless and Mobile Computing, Networking and Communications.

[2] Daniela Córdova-Pintado, Adrián Sánchez-Carmona and Ramon Martí, **“Interest-based Routing in Opportunistic Networks: Evaluating IRP against SCORP”** in 2023, 19th International Conference on Wireless and Mobile Computing, Networking and Communications.

Additionally, we wrote two more articles that have been submitted to two JCR journals, which at the time of presenting this thesis, are under review:

[3] Daniela Córdova-Pintado, Adrián Sánchez-Carmona and Ramon Martí, **“Interest-based Routing (IRP): An efficient routing strategy for receiving messages based on users’ interests in OppNets”** submitted to IEEE Transactions on Mobile Computing Journal (under review since January 2024).

[4] Daniela Córdova-Pintado, Adrián Sánchez-Carmona and Ramon Martí, **"Redefining Metrics for an Efficient Interest-based Routing in Opportunistic Networks"** submitted to Computer Networks Journal (under review since December 2023).

Related Work

2

In this chapter, we present the Related Work about our research. Firstly, we present a review about some of the existing opportunistic routing strategies and, a qualitative comparison with their support to the required characteristics in scenarios where destinations change in real time. And then, we also present a review about some existing performance metrics used to evaluate the routing algorithms.

One of the aims of this Related Work is to determine whether the routing strategies presented can be used to achieve the defined objectives. And, the other one is to show that for certain scenarios, traditional metrics such as Delivery Ratio and Message Latency are not so accurate, and therefore they must be redefined.

2.1 Opportunistic Networks

Opportunistic Networks (*OppNets*) [1] are a type of network where there is no fixed infrastructure and in which contacts between mobile devices are exploited for message forwarding. Contacts between nodes in these networks are intermittent, and there is no fixed path from source to destination of a message. So, in this type of network, the *store, carry and forward* paradigm must be used to reach the message destination.

In the past, *OppNets* have been referred to as pocket-switched networks [18] and also as people-centric networks [7]. In fact, they are sometimes considered like a special type of mobile ad hoc networks (MANETs) in which people carries mobile devices that communicate directly between them via some short-range wireless technology such as Wi-Fi or Bluetooth.

Nowadays, different kind of applications like space communication, wildlife monitoring, social applications, cellular traffic offloading, vehicular networks and challenged environments [30] can make use of *OppNets*. But in order to make them work in this type of networks, routing strategies with characteristics adapted to their specific needs are required. Therefore, it is important to continue developing new routing strategies that fit the needs of such applications and contribute to the creation of new ones.

2.2 Opportunistic Routing Strategies

This section presents a review of some routing algorithms in Opportunistic Networks (*OppNets*). Some of them flood the network using copy-based strategies. Others use utility-based strategies that take advantage of the encounters' history, multicast strategies that try to deliver messages to a group of destinations, or exploit the social characteristics of the nodes to take smart routing decisions. Finally, at the end of the section, a summary of the presented strategies and their support for our required characteristics is presented.

2.2.1 Copy-based routing

First, we analyze the Copy-based routing strategies, that are the ones that generate a certain number of message copies for the distribution of messages. Among them, the most basic one is the Epidemic (*EP*) [42] algorithm, which, in order to reach the final destination, forwards a copy of the messages to all nodes contacted, flooding in this way the network. Although Epidemic tries to get the message to the destination as soon as possible, it is a huge waste of network resources.

Another known algorithm is Spray & Wait (*S&W*) [39], that is based on flooding the network with a limited number of copies of each message, and waiting until one of the nodes with a copy of the message contacts the destination.

Other strategies have also been implemented, like [22, 21, 24, 4]. All of them include an epidemic-type distribution of messages, and they usually sent those messages to a fixed and known destination.

2.2.2 Historic Encounter-based routing

There are also utility-based strategies that use the history of encounters between nodes together with some of the node's characteristics (such as battery charge level, current location or movement information) to route messages to their destinations.

These approaches, like PRoPHET [26], reduce flooding by exploiting the likelihood of real-world encounters and maintaining a set of probabilities for delivering messages to fixed and known destinations.

A Multi-copy routing algorithm named iPRoPHET (improved PRoPHET) [40], which is an enhanced version of PRoPHET, covers some of its limitations by using a machine learning classifier called random forest to improve the delivery probabilities of PRoPHET to establish the route of the messages to their destinations. So, with PRoPHET messages are forwarded to nodes with higher delivery predictability to reach the established destination, while with iPRoPHET messages can also be forwarded to nodes with lesser PRoPHET probability but with other better contextual attributes.

Authors in [3] propose to deliver messages by detecting communities and node interest. The community detection process is based on the K-clique algorithm, which identifies sets of nodes with more frequent interactions. Nodes that share interests exchange message summaries and community sets, deciding to send messages based on interest coincidence. This exchange is based on each node having a previously known list of its peers, which is used to select receiving nodes that belong to predefined communities and are interested in the messages.

In [2], the authors propose a Dynamic Resource-Aware routing, where the message forwarding decision is based on an estimation of the time that a message will take to reach its destination.

The proposal presented in [14] is a message forwarding strategy that evaluates node quality and uses thresholds to make forwarding decisions. Node quality is calculated based on metrics such as connectivity, buffer capacity, and available energy. During a contact, a node adjusts its threshold based on the perceived quality of the encountered node, and only forwards the message if the other node has a higher quality than any other node seen so far by this message.

A strategy that uses node mobility patterns to make data forwarding decisions is presented in [32]. Precise attempts to predict the future location of nodes in a network, and make decisions based on these predictions. With Precise, nodes exchange the missing messages with their peers when they are in communication range and within a Replication Zones (RZ). Exchanges outside the RZ are performed only if a node has a high probability of reaching another RZ before a time threshold expires. When nodes are not expected to visit the RZ's, it assumes the data are not relevant for them.

All of these routing strategies require previous knowledge about the destination of the messages and keep data about the nodes they encounter and their relevant information in order to send the messages to them and to calculate the probabilities of encounter.

2.2.3 Multicast-based routing

Multicast-based routing strategies are an active research area in OppNets that try to enable the delivery of messages to multiple destinations that belong to a specific multicast group.

A weighted network contact graph, which reflects the node importance and contact tightness between nodes is described for building a multicast backbone that is used to forward the messages to their destinations, is presented in [46]. A couple of Multicast-based routing methods are also proposed in [16], that are based on the relays' cumulative probabilities in order to forward single or multiple data to fixed destinations.

In [10], the Social Profile-based Multicast (SPM) routing scheme is presented, supporting an efficient multicast message broadcasting based on social profile. A set of important and representative static social features is identified (affiliation and language). Then, these features are used as routing metrics to forward the messages to their destinations (a group of nodes). Also, SPM-Overhead Reducing (SPMOR) is presented in the same paper, as an extension to SPM. Taking into account the time-varying social behaviors during a daytime and nighttime, it restricts the number of relays during the daytime, that is when there are more contacts and a few numbers of forwardings is enough to achieve a desirable delivery ratio.

An Encounter-based multicast routing (EBMR) scheme based on PRoPHET is presented in [44], where it uses a wait timer (WT) to decide to use PRoPHET together with a delivery threshold (if the timer has not expired) or to deliver it to any node it encounters (when the timer has expired). Messages are sent to a list of known destinations, and nodes can forward the messages to subsets of these destinations (only one copy for each destination) according to the delivery probabilities of the contacted nodes.

Although the Multicast-based routing algorithms may be effective in certain network contexts, they currently do not support scenarios where users can dynamically create topics and the message routing is based on user interest or message duration is determined by interest.

2.2.4 Social Characteristics-based routing

The use of social characteristics has emerged as a new opportunity to improve the delivery of messages on *OppNets*. Some studies [47, 5] present reviews on social-aware routing protocols or on routing protocols based on social relationships.

In [17], the authors study the behaviors of user data access over different categories of web data. They reveal that taking into account the users' interest in Opportunistic Networks (*OppNets*) is a promising strategy for the future of this kind of networks.

Authors in [28] describe a forwarding mechanism, where the cosine similarity metric between the interest profiles of the nodes, and the total time a pair of nodes are in contact are used for making forwarding decisions. In it, nodes must previously declare their interest, and it assumes that individuals with similar interests tend to meet more often.

In [43], authors propose ICON, a platform that manages the location and digests of user interests. It is an intermediate, deployable solution that supports *OppNets* in urban areas. However, it relies on a centralized server and the usage of mobile data plans.

In the proposed ICast [31], nodes use a counter to keep track of the number of encountered nodes that are interested in every topic, and forward the messages towards the nodes with higher values than its own. After forwarding a message, nodes decide between dropping it or continue carrying it based on their own interest in the topic.

The protocol presented in [19] applies social network analysis and exploits social and structural metrics to forward messages to their fixed and known destination. Another one, [45], uses the degree of interest similarity to identify potential subscribers. It calculates a utility value, taking into account social characteristics and residual energy, to determine suitable relay nodes. Also, it calculates the number of replicas of every message that should be forwarded based on the centrality degree and the residual energy.

Onside routing algorithm, presented in [6], uses the node's online social connections (friends on social networks), common interests, and contact history as the mechanism for deciding to forward messages to their destinations. Additionally, in [41], data popularity is used for prioritizing messages and forwarding them in the network to their destinations.

Social-aware Content-based Opportunistic Routing Protocol (*SCORP*) [29] uses a utility function that reflects the probability of encountering nodes with a certain interest for forwarding the messages to their destinations. These nodes are the best socially connected nodes that learn about levels of social interactions by using the time they stay connected to other nodes with similar interests. Regarding the forwarding decision, for each contact, *SCORP* compares the social weight of interest of both nodes and only forwards the message if the other node is interested in the message, or if it has a higher weight than it does. It uses the FIFO (First In First Out) policy when it has a full buffer and has to decide which message to discard. Finally, in *SCORP*, the authors declare its advantage for using messages with a timely, limited utility.

Although some of these strategies use interests as part of their proposals, neither of them contemplates that users do not necessarily must have common friends or interests to relay a message, or that this interest can change according to users' interest at any time, and, therefore, the messages destinations also change. Also, neither of them contemplates that messages can remain on the network as long as users are interested in their topics without necessarily removing them using the TTL.

2.2.5 Summary of presented strategies and required characteristics

In order to obtain a pure OppNet solution that allows applications to efficiently send messages on any topic to multiple destinations that can change at any time, some characteristics are required, which are presented below:

Users should be able to create messages on any topic at any time and to send them without knowing who will receive the messages.

Also, users should be able to decide whether they want to receive messages on any topic at any moment, therefore, messages' destinations must change in real time according to users' choices.

Users should receive messages that are of interest to them, regardless of the time of message creation.

Finally, users' interests must be used to route messages through the network and to decide if they should be dropped.

In subsections 2.2.1 to 2.2.4 we have analyzed and discussed some existing *OppNets* routing strategies. Table 2.1 provides a summary of them and shows their support, or lack of it, to each of the required characteristics in order to allow the development of applications that efficiently transmit messages.

Some of the Copy-based routing strategies support multi-destination messages to fixed and known destinations. The Historic Encounter-based routing strategies do not support any of the required characteristics. The Copy-based are inefficient in

Characteristics	Routing strategies			
	Copy-based	Historic Encounter-based	Multicast-based	Social Characteristics-based
Dynamic topics creation by user	×	×	×	○
Multiple destinations for every message	○	×	✓	✓
Message's destination is decided by receiver	×	×	✓	✓
Message's destination change in real-time	×	×	✓	✓
Messages routed based on user's interest	×	×	×	○
Message lifespan based on interest	×	×	×	×

✓Supported × Not supported ○ Supported by some of them

Tab. 2.1: Summary of the discussed routing strategies and their support of the required characteristics

their use of resources, filling the network and buffers with packets regardless of the user's interest. Alternatively, Historic Encounter-based requires tracking the existing nodes, to calculate encounter probabilities and predict future node locations, as well as previous knowledge about message destinations for message delivery.

The discussed routing strategies grouped both in Social Characteristic-based and in Multicast-based provide support to multi-destination messages, allow destinations to change in real-time, and allow the users to decide by themselves if they want to be the destination. But, none of these strategies support defining the message lifespan based on the interest of the nodes.

Some Social Characteristic-based routing strategies, the ones presented in [29, 43, 31, 6, 41] also make routing decisions based on user's interest, and one of them [29] even allows users to create topics dynamically. Anyway, it must be noted that these characteristics are not usually supported by the majority of Social Characteristic-based routing.

However, none of them supports all the required characteristics together. So, in order to allow applications to efficiently use the resources for delivering messages based on the user's changing interests to multiple destinations, it would be helpful to provide a new strategy that supports all of these characteristics.

2.3 Performance evaluation metrics

For all kind of routing protocols, including the *OppNets* ones, it is important for the researchers to be able to evaluate their performance and compare them with others. This subsection presents a small discussion of the usual methods used to evaluate opportunistic algorithms.

In order to evaluate routing protocols, researchers tend to use traditional metrics such as Delivery Ratio or Message Latency [38] to evaluate their *OppNets* routing proposals and compare them to other ones.

Analyzing the algorithms in Section 2.2, we see that all of them use these traditional metrics for evaluation and performance comparison purposes. But that is not always the case, and other researchers have needed to adapt traditional metrics due to the characteristics of their proposals [33, 35].

The authors of [33] propose to consider spatial and temporal aspects when calculating Delivery Ratio and Latency for geocasting. They suggest assigning a delivery ratio to each message, which is the division of the number of devices that received the message between the total number of devices that were located in the cast area throughout the lifetime of the message (so they should have received the message). Then, the overall Delivery Ratio is the result of averaging these ratios. On Message Latency, authors measure the per-message latency as the time it takes for a message to reach a device in the destination cast. The overall Latency is the average of all messages' latency.

Another case where authors consider that the traditional metrics do not provide good enough information can be found in [35]. In it, the authors present LEPTON, a new emulator for Opportunistic Networks (*OppNets*), that also includes an alternative form of calculating the delivery ratio. They propose to calculate it as the ratio between the number of effective receivers and the *ideal* number of receivers, as predicted by the horizon computation included in the LEPTON toolkit.

The horizon of a message is the set of all potential receivers for that message, considering all contacts between nodes after its creation, and it is calculated after modelling the whole network and all contacts as a dynamic graph.

After this brief review about metrics, it should be considered that sometimes, due to the characteristics of an application or a specific scenario, traditional metrics do not provide good enough information about the performance of the algorithms. Therefore, sometimes it is necessary to develop new metrics or to adapt them, as in the case when there are multiple destinations that change in real time according to the changing interests of the users, which we propose.

Part II

Contributions

Interest-based Routing

3

WHEN message destinations change in real time according to the users' interest in a topic, is necessary a routing strategy that adapts efficiently for delivering as many messages as possible. For this reason, we designed Interest-based Routing Protocol (*IRP*), the new routing strategy that focuses on leveraging the users' interest to improve message dissemination in *OppNets*. Therefore, in this chapter we present *IRP*.

We explain the design of *IRP* and each one of its components and how they operate in order to accomplish the defined objectives. The two fundamental concepts of "*Interest in a topic*" and "*Perception of interest in a topic*" are introduced in order to show how they are combined to make efficient forwarding decisions by prioritizing the topics of higher global interest in the network.

Finally, we show how *IRP* manages its buffer to maximize the use of network resources and allow messages to persist in the network as long as there is interest in them, thus not depending on the use of TTL to discard messages.

3.1 Interest-based Routing Protocol *IRP*

The Interest-based Routing is a new strategy to support the implementation of applications where network users interact to efficiently send messages on different topics to other unknown interested users, who can change their interests at any time. So, this new routing solution aims to:

- Allow real-time changes in users' interests on topics, and therefore adapt to the fact that message destinations can change at any time.
- Allow users to define and create message topics dynamically, so they can use them at will without any restriction on the definition of the topics or the moment to send the messages.
- Deliver as many messages as possible on the topics in which there is a high global interest.

- Improve the use of network resources by allowing messages to remain on the network as long as there is interest in them, while preventing the network from being flooded with unnecessary messages when there is not enough interest in them.

In order to achieve the previous objectives, we propose the Interest-based Routing Protocol (*IRP*), a new routing solution based on user's interests that use the concepts of *Interest in a topic* and *Perception of Interest in a topic* for making forwarding decisions. This strategy considers that the interest in receiving messages on different topics can change in real-time, so message destinations also change at any time.

The *Interest in a topic* is defined as the measure of how interested a user is in receiving messages regarding a specific topic, while *Perception of Interest in a topic* is defined as the measure of the *Interest in a topic* that nodes perceive around them regarding every topic, based on their encounters with other nodes.

Figure 3.1 shows how we want our proposal to operate:

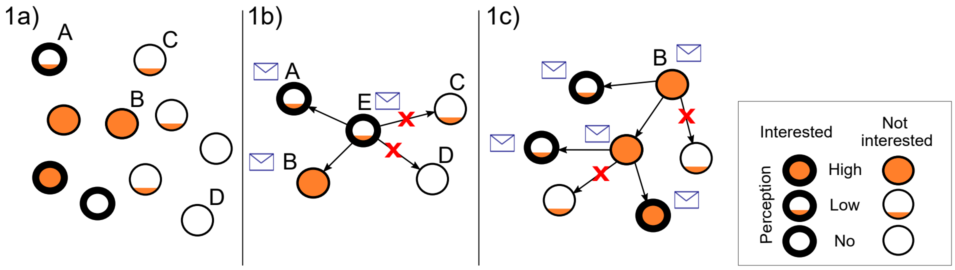


Fig. 3.1: Interest-based operation. 1a) Node A is interested and nodes B, C and D are not; Node B perceives a high interest among its neighbors while nodes A, C and D perceive a low interest. 1b) Node E encounters A, B, C and D. It forwards the message it carries to A (it is interested) and B (it has a high perception), but not to C or D (they are not interested and have low or no perception). 1c) Later, node B contacts other nodes and forwards the message to those that have a high perception or are interested.

1a) At any moment, nodes may (or may not) be interested in a topic and also perceive how much interest in that topic is around them. Interested nodes like A want to receive messages from this topic, nodes like B perceive a high interest in the topic among their neighbors and could forward messages to them, nodes like C and D are not interested and perceive a low or no interest around them.

1b) Nodes A, B, C and D move and meet a node E that is carrying a message. Since A is interested in the topic of the message, and B perceives that there is high interest around it, E forwards the message to A and B, but not to C and D, because they are not interested and do not see enough interest around them.

1c) Finally, the node B, that is not interested in the message but carries it for the sake of their neighbors, finds other interested and high perception nodes and forwards the message to them, that later forward it to other interested and high perception nodes.

3.2 User Interest in a topic

One of the key concepts in the proposal is the interest that users have in receiving a message on a specific topic. This interest is modelled as the *Interest in a topic*, and it is used as part of the message forwarding strategy to decide whether to forward messages.

In the proposal, all messages are associated with one specific topic. When a user generates a message, this user must decide on a topic tag (t) to attach it to. Then, *IRP* tries to deliver the message to all users that are interested in that topic.

Each user operating a node N has different interests in different topics. We define a user's interest in receiving messages on a specific topic, t, as their *Interest in a topic*, represented by an integer value, denoted as I_t^N. The user can set I_t^N to 1 or 0 according to its own preferences. 1 means that the user is interested in receiving messages of topic t, and 0 implies that the user is not. Users can change their interest I_t^N in real-time whenever they decide to do it.

3.3 Perception of Interest in a topic

The other key concept that *IRP* uses is *Perception of Interest in a topic*, which models the node's perception of interest in a topic around it. This section explains how this perception is built when nodes contact each other.

The *Perception of Interest in a topic*, P_t^N, defines how the node N perceives the interest that exists in the network in the topic t. Nodes should have a high P_t^N when they frequently contact other nodes that are interested or that perceive a high interest in t. On the contrary, nodes should have a low P_t^N when they frequently contact other nodes that are not interested and perceive a low interest in t.

Figure 3.2 shows how nodes around interested nodes that usually frequent specific locations form small areas of high perception. *IRP* will use the nodes in these areas to forward the messages to the nodes that are interested in them. Meanwhile, nodes frequently surrounded by not interested nodes create areas of low or no perception, and *IRP* will benefit from this in order to avoid wasting resources by forwarding messages that nobody around seems to be interested in receiving.

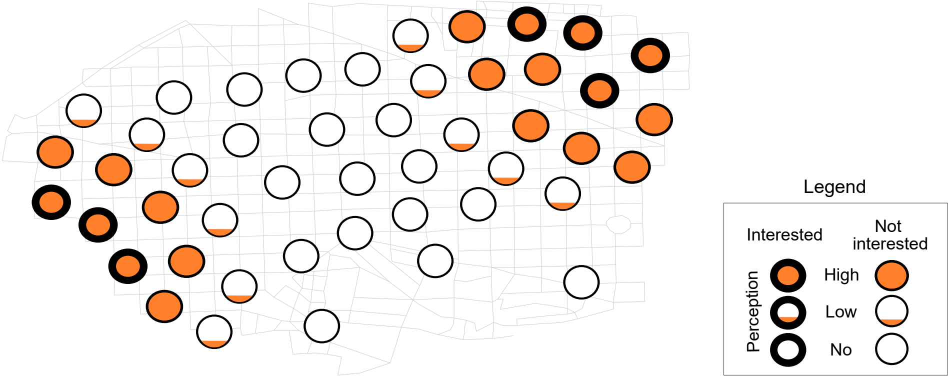


Fig. 3.2: Areas of interest and perception. Interested nodes that frequent a certain location form around them areas of high perception nodes, while the nodes that are not interested and usually only meet other nodes that are not interested and have no perception form areas of low perception nodes.

In order to calculate its *Perception of Interest in a topic* (P_t^N), the node N initializes P_t^N using its value of I_t^N. From that moment on, P_t^N is updated each time N encounters another node. This update is calculated using two Exponentially Weighted Moving Average (EWMA) equations, as we explain below.

Every time a node A contacts[1] another node B, A updates its P_t^A, for every topic t known by A or by B. In order to update P_t^A, the node B calculates a *View on a topic*, V_t^B, which is a measure about how interesting B considers this topic is to the network and to itself, using its own I_t^B and P_t^B, as shown in Equation 3.1. In case one node does not know one of the topics in the other's topic list, before making the calculations, it creates and initializes a new $I_t^N = \text{false}$ and $P_t^N = 0$ for this new topic, thus propagating it through the network.

$$V_t^B = I_t^B \cdot (1 - \beta) + P_t^B \cdot \beta \tag{3.1}$$

where,

$\beta =$ Weight given to the *Perception of Interest in a topic* of a node B, in relation to the *Interest in a topic* of the same node, $\beta \in [0, 1]$.

The β parameter controls the impact that the *Interest* and the *Perception of Interest in a topic* of the encountered nodes have on a node's perception. A high β gives more weight to the perception of the interest of the neighbor nodes than to its own

---

[1] Implementation note: nodes can send their *Views on a topic* and the list of topics they are interested in during the discovery phase in order to accelerate the forwarding decision.

interest. On the other hand, a small β gives more weight to the own interest than to the perception of the found nodes.

Then, A uses the *View on a topic* from B, V_t^B, to update its P_t^A on topic t, using Equation 3.2.

$$P_t^A = P_t^A(old) \cdot (1 - \alpha) + V_t^B \cdot \alpha \tag{3.2}$$

where,

$P_t^A(old)$ = *Perception of Interest in a topic* of node A on topic t before node A contacted node B.
α = Weight given to the *View on a topic* of node B in relation to the previous perception of node A, $\alpha \in [0, 1]$.

The α parameter allows controlling how the encounters' history is weighted when updating P_t^A. A high α gives more weight to the most recent encounters, so nodes adapt their perception quickly to changes of interest in the network. On the other hand, a small α gives more weight to older encounters, allowing the node's perception to be less influenced by currently acquired information. This way, nodes are slower to perceive changes in the overall network interest.

It must be noticed that simultaneously to these calculations done by node A, node B also updates its P_t^B using V_t^A.

Figure 3.3 shows the evolution of a node's perception when it moves from one location to another. Node A's perception increases or decreases according to its encounters with interested and high perception nodes or with not interested and low perception nodes.

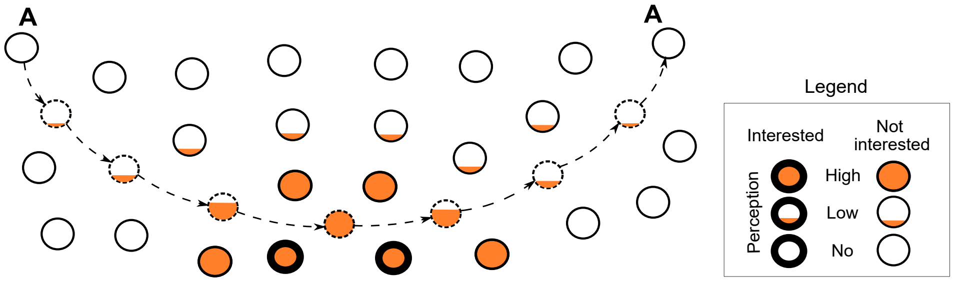


Fig. 3.3: Perception's evolution. Node A is not interested and begins with no perception of interest on a certain topic. While it travels, its perception of interest evolves according to its encounters with interested and high perception nodes (making its own perception rise) or not interested and low perception nodes (making its own perception decrease).

Finally, we designed Interest-based Routing on the assumption that all users in the network are trustworthy. So, their behavior is according to the requirements of our algorithm when they interact, compute and exchange the view, the interest, and the perception. We also assume that there are no risks that compromise privacy, integrity, and confidentiality when exchanging messages.

3.4 Forwarding threshold

In order to determine if there is enough interest in the network to forward the messages on a certain topic, we have defined the *Forwarding threshold*, T_f, which is a global value shared between all nodes in the network.

The choice of T_f directly influences the forwarding decision. Using a low T_f, nodes will forward more messages, and the risk of network flooding increases when lots of nodes are interested in some topics. However, when there are few interested nodes the possibilities of reaching more destinations increases. It must be noticed that using $T_f = 0$, *IRP* behaves like Epidemic. On the other hand, using a high T_f, nodes will forward fewer messages and limit the network flooding, but there is a risk of delivering fewer messages when there are not enough interested nodes. In the case of using $T_f = 1$, *IRP* only provides direct delivery to interested nodes.

3.5 Forwarding decision

Whenever node A contacts node B, A must decide, for every message m, whether to forward it. In our proposal, first, A checks the messages that B has already received, so only messages that B does not have will be forwarded.

Then, if B is interested in one or more topics, A will forward to B the messages of these topics. But in the case that B is not interested, *IRP* uses the *Perception of Interest in a topic* as the forwarding decision mechanism, comparing it to the *Forwarding threshold*, T_f. So, the messages of topics with a P_t^B higher or equal to this T_f will also be forwarded. Node A forwards the messages in the following order:

1. Firstly, A forwards the messages on topics that B is interested in receiving ($I_t^B = 1$) sorted by B perception, P_t^B, (highest to lowest). When two or more messages have the same perception, the most recently received messages are forwarded first.

Algorithm 1 *IRP*'s Forwarding

```
begin when A contacts B
for each ($m \in$ A.buffer) do
    $t \leftarrow$ m.GetTopic()
    if $m \notin$ B.buffer and [$(I_t^B = 1)$ or $(P_t^B \geq T_f)$] then
        A.AddToOutgoingMessages($m$)
    end if
end for
A.SortAndSendOutgoingMessages(B)
A.UpdatePerceptions(B)
end
```

2. Secondly, A forwards the messages on topics in which B is not interested but $P_t^B \geq T_f$, also sorted by perception (highest to lowest). When two or more messages have the same P_t^B, the most recently received messages are forwarded first.

This forwarding decision, explained in the previous paragraphs, is summarized in Algorithm 1.

Figure 3.4 illustrates the forwarding decision. Messages on topics that node B is interested in receiving are sorted by B's perception from highest to lowest (e.g., messages received in time 5, 6 and 9). These messages are the first to be delivered. Then, messages in which the node B is not interested but has a perception higher or equal than T_f are also forwarded (e.g., messages received in time 1 and 3), sorted by B's perception from highest to lowest. In all cases, messages on topics with the same perception are forwarded, sorted by their age (newer first). Note that messages of no interest to B, and that B has a low perception of its topic (message received in time 8) are not forwarded.

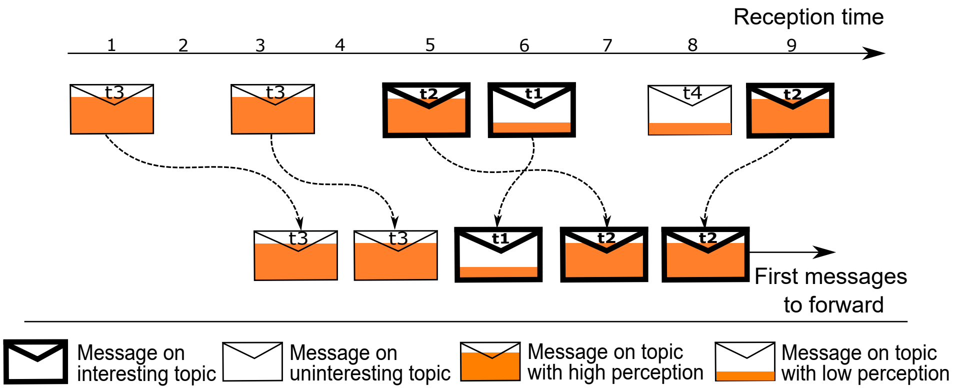


Fig. 3.4: Forwarding decision. Nodes first forward the messages on topics of interest to the other node, sorted from highest to lowest perception, and then, messages on topics with high perception, also sorted from highest to lowest perception. In both cases, messages on topics with the same perception are sorted by their age (newer first).

3.6 Buffer management

When a node has its buffer full, and it is about to receive a new message, it must decide which message to drop. Our *Buffer management* proposal tries to improve the overall performance by keeping the messages of topics with higher interest in the network. In order to do that, we consider two criteria for dropping messages:

1. The *Perception of Interest in a topic*, P_t^N
2. The time in which each message was received.

Using the criteria listed above, the node drops the oldest message received from the topic with the lowest P_t^N. Figure 3.5 shows how the buffer works when nodes need to drop messages.

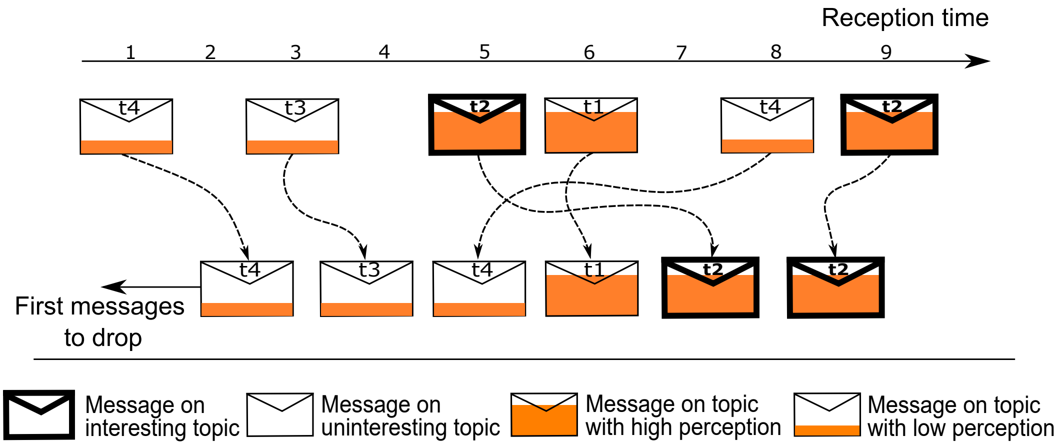


Fig. 3.5: Buffer management. Nodes drop messages when they are about to receive a new message and there is not enough space in the buffer. The message dropped is always the oldest message received of the topic (or topics) with the lowest perception.

Messages on topics $t4$ and $t3$ have the same lowest P_t^N, so these messages are sorted from the oldest received to the most recent; in this case, the message on $t4$ received at time 1 is the oldest received with the lowest P_t^N, therefore, it will be the first to be dropped. Then, every time the node needs to drop a message, the same process of message sorting is repeated, and the oldest message with the lowest P_t^N is dropped again (message on $t3$, then $t4$, and so on).

With this proposal, messages are removed from the buffer, and thus from the network, when nodes perceive no more interest in them, leaving space for newer and more interesting messages. Therefore, *IRP* does not need to use the Time To Live (*TTL*) field in messages. This way, *IRP* avoids the arbitrary deletion of messages that could still be interesting and allows them to be delivered long before their creation, as long as there is interest in the network.

Note that security considerations in terms of confidentiality, privacy, and integrity are out of the scope of this work. However, in order to address this, a general security framework like [23] or [25] can be used, as *IRP* is compatible with them.

In summary, in this chapter, we presented our first Contribution, the new routing protocol for Opportunistic Networks, Interest-based Routing Protocol (*IRP*), that uses *Interest in a topic* and *Perception of Interest in a topic* for making forwarding decisions. We also presented a novel *Buffer management* in which messages are discarded when nodes perceive that there is no interest in their neighborhood, instead of using the *TTL* that removes messages only considering their creation time.

Redefined Metrics based on users' interest

4

WHEN we have scenarios where users' interests change in real time and the destination of messages also change accordingly, traditional metrics such as Delivery Ratio or Latency do not provide good enough information about the performance of different algorithms. Considering these facts, we redefined these metrics to allow the analysis of algorithm performance by taking into account only the time when messages can be delivered. Therefore, in this chapter, we present the redefinition of Delivery Ratio and Latency.

Throughout this chapter, we provide an explanation about how to calculate the Delivery Ratio taking into account the nodes that were interested in receiving a message, and how to calculate the Latency considering only the periods of time in which the nodes are interested in a topic.

4.1 Delivery Ratio

Traditionally, when analyzing the Delivery Ratio in routing algorithms, it is calculated as the relation between the number of messages delivered to all nodes, m_d, and the potential number of messages that could be delivered, m_{pd}, as expressed in Equation 4.1.

$$DR = \frac{m_d}{m_{pd}} \tag{4.1}$$

When there is only a single destination for each message, m_{pd} is the number of all messages that have been sent. In the case that the messages have multiple destinations and this number is the same for all messages, m_{pd}, is calculated as the number of all messages that have been sent multiplied by the number of possible destinations. On the other hand, if the message has multiple destinations, but this number is different for each message, m_{pd}, is calculated as the sum of all possible destinations of each message.

Usually, traditional routing algorithms send messages to a specific node or set of nodes. This way, Equation 4.1 is used, and it is easy to calculate the Delivery Ratio. Typically, these algorithms do not intend to send messages to a number of destinations that may change in real-time based on user needs. Therefore, with *IRP*, calculating the Delivery Ratio in the usual way becomes a problem for the following reasons:

1. The nodes can switch between being interested and not interested in a topic at any time, changing who should receive every message on that topic.
2. Messages can be delivered to multiple nodes.
3. So, for every message, there is not an immediate way to know how many nodes, and which ones, should receive every message.

Since in *IRP* the number of destinations changes in real-time according to the user's interests, it is not straightforward to calculate the Delivery Ratio in the traditional way. Moreover, when nodes become disinterested, they do not want to receive any more future messages on that topic, but still need to be accounted for the messages they received and the ones they could have received while they were interested. Consequently, we propose calculating the Delivery Ratio, DR', taking into account these two considerations.

The number of delivered messages on every topic, m_d^t, remains unchanged. However, the number of potential messages that could be delivered in every topic (m_{pd}^t) cannot be neither the number of messages sent, as there are no single destinations for the messages, nor the number of messages sent multiplied by the number of potential destinations, because these are changing in real time based on the users' needs. We make a distinction between the potential number of messages that could be delivered to nodes currently interested in a topic, $m_{pd}^t(int)$, and the potential number of messages that could have been delivered to nodes that were interested at some point, but they are no longer, $m_{pd}^t(\neg int)$; therefore, we express m_{pd}^t as shown in Equation 4.2:

$$m_{pd}^t = m_{pd}^t(int) + m_{pd}^t(\neg int). \tag{4.2}$$

The number of messages that could have been delivered to the nodes that are still interested in a topic, $m_{pd}^t(int)$, can be calculated as the product between the number of messages generated on the topic, m_g^t, and the number of nodes currently interested in that topic, $\#int$ (Equation 4.3).

$$m_{pd}^t(int) = m_g^t * (\#int) \tag{4.3}$$

Regarding the number of messages generated that could have been delivered to the nodes that are no longer interested in a topic, $m_{pd}^t(\neg int)$, it is calculated using Equation 4.4.

$$m_{pd}^t(\neg int) = \sum_{N=0}^{\#nodes} m_g^t(time_{N\neg}) \tag{4.4}$$

We define $m_g^t(time_{N\neg})$ as the number of messages that were generated in topic t until the moment when the node N became disinterested in it, and we calculate it for every node in the network that was interested in topic t at some moment (note that $m_g^t(time_{N\neg}) = 0$ for every node N that has never been interested in topic t). This way, messages created after the node lost interest in a topic are not taken into account because they could never be delivered.

Finally, using the defined variables explained above, the equation for calculating the Delivery Ratio is expressed in Equation 4.5.

$$DR' = \frac{m_d^t}{m_{pd}^t(int)+m_{pd}^t(\neg int)} = \frac{m_d^t}{m_g^t*(\#int)+\sum_{N=0}^{\#nodes} m_g^t(time_{N\neg})} \tag{4.5}$$

It is important to highlight that this is the Delivery Ratio equation that we use when analyzing the results obtained in Chapter 7.

4.2 Messages Latency

When analyzing traditional algorithms, the Latency of a message m (L_m) is calculated as the time elapsed between the moment when the message was generated (T_{m_g}) and the moment when it is delivered (T_{m_d}), as expressed in Equation 4.6.

$$L_m = T_{m_d} - T_{m_g} \tag{4.6}$$

Calculating the Latency in this way, it is assumed that nodes are always willing to receive messages, so using equation 4.6 is enough. However, this approach does not take into account that the user may lose interest during a period of time, and that, during that time, the message can not be delivered to the node even if a contact has it or if the node carries it in its own buffer.

Since in our scenario users change their interest in receiving messages in a topic in real-time according to their preferences, creating periods in which they do not want to receive any message, it is not fair to calculate Latency in the traditional way.

Consequently, we propose to calculate the Latency, L'_m, taking into account these considerations.

When L_m is calculated, it takes into account all the time elapsed between the message generation and the message delivery, but when considering the interest of the node, in L'_m, the time between consecutive periods of interest is discounted, because in that periods the nodes are not interested and the messages cannot be delivered.

Figure 4.1 shows the difference between the traditional Latency (L_m) and the Latency that we propose (L'_m), considering a node that was interested in receiving messages during five different periods.

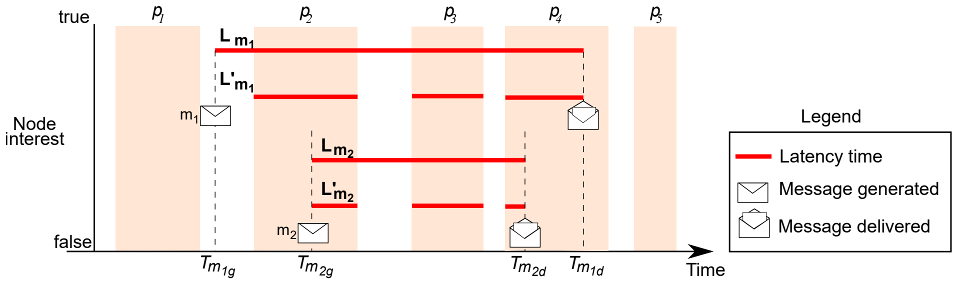


Fig. 4.1: Traditional Latency (L_m) Vs Proposed Latency (L'_m). Note that L' omits the periods of time in which the node is not interested in the message topic. L'_{m_1} is counted from the moment in which a node starts to be interested in a topic, because the message was generated out of a period of interest, while L'_{m_2} is counted from the moment when the message is generated because it was generated in a period of interest.

Now, in order to calculate L'_m, we only take into account the periods p elapsed since a message is generated ($T_{I_t^N}(end)_p > T_{m_g}$) and until the message is delivered ($T_{m_d} > T_{I_t^N}(init)_p$) as in periods p_2, p_3 and p_4 in Figure 4.1. So, periods ending ($T_{I_t^N}(end)_p$) before T_{m_g} or starting ($T_{I_t^N}(init)_p$) after T_{m_d} are not counted in L'_m.

Besides, a message can be generated before a period of interest, as message m_1 in period p_2 (but also p_3 and p_4) in Figure 4.1, and in this case the contribution of the period to L'_m is counted from the moment in which they start ($T_{I_t^N}(init)_p$). But the message can also be generated within a period of interest, as m_2 in period p_2, and in this case the contribution to L'_m is counted from T_{m_g}.

Furthermore, a message can be delivered within the period of interest we are studying (p_4 in Figure 4.1), and then L'_m is counted until T_{m_d}. But it can also be delivered at a later period (this happens to p_2 and p_3 in Figure 4.1), and then the contribution of that period to L'_m is counted until the period ends ($T_{I_t^N}(end)_p$).

With all the previous considerations, we define P_t as the set of all periods (p) in which a node has been interested in a topic t. Then, let $\bar{P}_t$ be the set containing all periods $p \in P_t$ such that they satisfy $T_{I_t^N}(init)_p < T_{m_d}$ and $T_{I_t^N}(end)_p > T_{m_g}$. Therefore, we calculate the Latency (L'_m) using Equation 4.7.

$$L'_m = \sum_{p \in \bar{P}_t} min\{T_{I_t^N}(end)_p, T_{m_d}\} - max\{T_{I_t^N}(init)_p, T_{m_g}\} \tag{4.7}$$

This way of calculating Latency is more accurate than the traditional one in this type of scenarios with changing interests, and it will be used in Chapter 7 to analyze the results obtained.

Part III

Experimentation and Results: *IRP*'s performance

5

The performance evaluation begins

When a new routing protocol is designed and implemented, it is a good practice to analyze its performance and compare it with other ones. Therefore, in this chapter, we present the performance evaluation of Interest-based Routing Protocol (*IRP*), Spray & Wait (*S&W*) and Epidemic (*EP*) and the comparative between them.

We designed the experiments in a scenario where the destination of the messages change in real time according to the user's interest, and run a set of simulations.

Then, with the obtained results, we provide the performance comparative analysis between *IRP*, *S&W* and *EP*.

The experiments and results shown in this chapter were presented in the conference paper titled "Paving the way towards delivering messages based on user interests in OppNets" at the 18th International Conference on Wireless and Mobile Computing, Networking and Communications in 2022 [8].

5.1 Experiments

This section presents the experiments we have defined to evaluate *IRP*'s performance using a normal distribution of user interests. Then, it is presented and analyzed the average results obtained by three executions of simulations, comparing the Interest-base Routing protocol (*IRP*), Epidemic (*EP*) and Spray and Wait (*S&W*).

5.1.1 Map

For these experiments, we used the actual map of the Eixample district of Barcelona (Spain). The map is 1.60 km wide and 3.06 km high, with an effective area of 4.8 km^2 (Figure 5.1). We performed them using *The ONE* simulator[20] and ran three different simulations using different random seeds. Then, the average results were calculated.

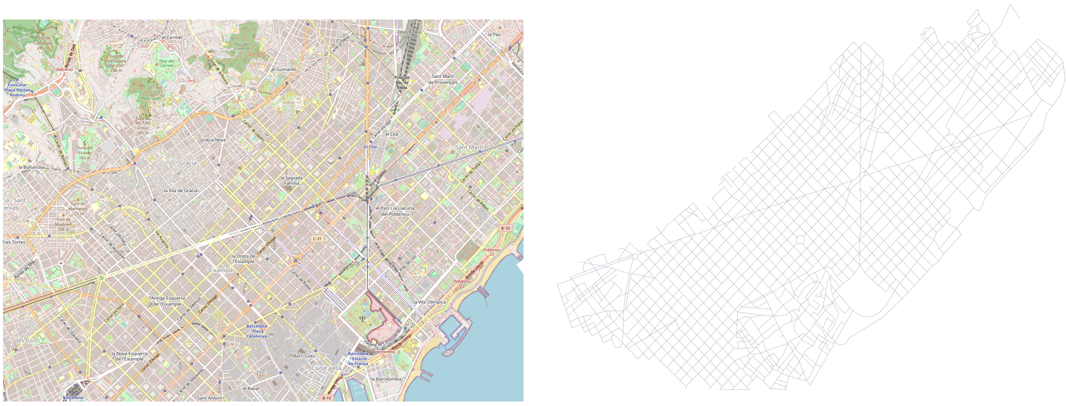


Fig. 5.1: Left: Map of the Eixample district in Barcelona used in our experiments. Right: Vectorial representation of the Eixample map used for the simulation in The One.

In order to evaluate *IRP*’s performance through simulations, since none of the cited algorithms support all the required characteristics together, the Epidemic (*EP*) and Spray and Wait (*S&W*) routing protocols have been chosen.

Although these protocols are maybe two of the most basic ones, they are a good choice for the analysis since many researchers use them for performance evaluations.

5.1.2 Mobility model

We define a synthetic mobility model where the nodes move with a map-based random movement pattern [37] in the map of the Eixample district. Although this is not a movement pattern that adjusts to daily life, this movement allows us to perform these first experiments and to test the efficiency of *IRP*.

5.1.3 Interest model

In these experiments, we model the amount of interested nodes in every topic (t) as a normal distribution.

We have defined four topics ($t0$ to $t3$) along 17 days. The number of interested nodes in each topic reaches a different maximum, between 30%, and 50% of the total number of nodes in the experiments. We decided to characterize the interests as shown in Figure 5.2 in order to cover all possible situations: between days 0 and

9, the interest in at least one topic is always present, and from day 0 to 6 there is a high interest in more than one topic at the same time; between days 9 and 11, there are around two days with no interest in any topic in order to analyze the behavior of the different protocols when this happens; finally, between days 12 and 16 we placed a high peak of interest in only one topic that has not been interesting before.

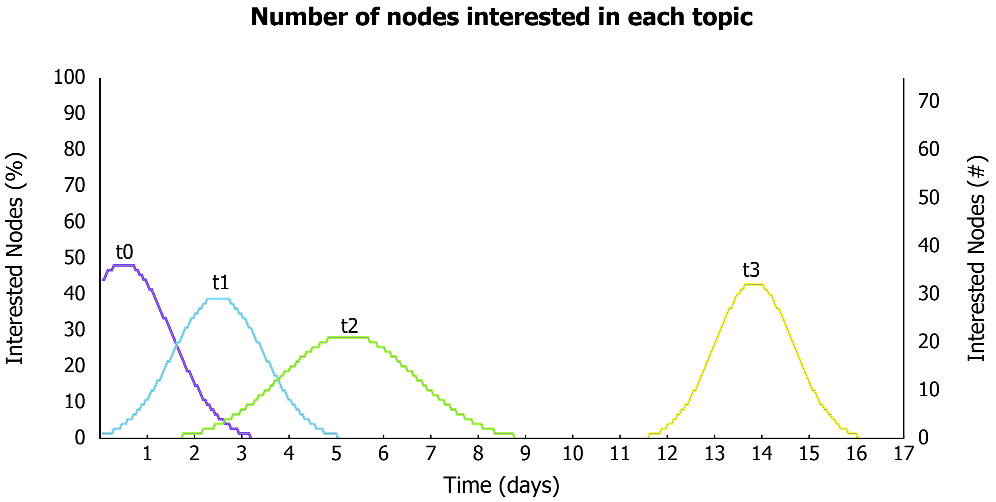


Fig. 5.2: Distribution of the number of nodes interested in four different topics during seventeen days of simulation.

The maximum number of interested nodes in a topic, and the duration of the interest (the time elapsed since the first node becomes interested until there are no more interested nodes), are shown in Table 5.1.

Topic	# Nodes	Duration of the interest
t0	37	80 hours
t1	30	120 hours
t2	22	160 hours
t3	33	96 hours

Tab. 5.1: Number of nodes interested and duration of the interest for each topic.

5.1.4 Experimental Settings

The parameters shown in Table 5.2 have been used to configure the set of simulations and to compare the performance of *IRP* with *EP* and *S&W*.

As previously explained, the values of α, β and T_f can be adjusted according to the needs of each case. In our case, we conducted many experiments using different values and selected those that provided the best results.

The value of L in *S&W* was selected according to the authors' suggestion [39] of using a number between 10% and 15% of all nodes used in the experiments. Additionally, we use an infinite *TTL* in order to demonstrate that the lifespan of the message depends on the interest in the topics of the messages in the network.

	Parameter	**Value**	**Units**
Scenario	simulation time	17	days
	number of simul.	3	simul.
Network	transmission range	10	m
	speed transmission	2	Mbps
Nodes	number of nodes	75	nodes
	speed range	3 - 50	km/h
	buffer size	10	MB
Messages	size	500-1000	kB
	speed of creation	60-72	msg/h
	TTL	∞	-
	topics	4	topics
IRP	α	0.3	-
	β	0.2	-
	T_f	0.7	-
Spray and Wait	L	7	msg

Tab. 5.2: Simulation parameters.

Also, we had to slightly adapt the behavior of Epidemic (*EP*) and Spray & Wait (*S&W*) to a scenario in which messages are not directed to nodes, but are delivered to them depending on their interest. In this sense, we made the following changes.

- When a message is forwarded to a node, it is analyzed whether its topic is one of the node's current interesting topics, and if it is, the message is considered as delivered to that node.
- When a node becomes interested in a topic, its buffer is scanned to find all messages of that topic. All these messages are considered as delivered. No messages are removed from the buffer when doing this.
- Moreover, the messages forwarded to a node are always added to its buffer, even if the message is considered as delivered. This way, all messages can be forwarded to other nodes.

5.2 Results

In this section, we present and analyze the average results obtained by five executions of the simulations in the experiments, comparing Interest-based routing protocol (*IRP*), Spray & Wait (*S&W*) and Epidemic (*EP*). For this analysis, we study the Messages Delivered, Messages Relayed, Buffer Occupancy, Messages Latency, Efficiency.

5.2.1 Messages Delivered

In order to compare *IRP* with the two other protocols, we first analyze the total number of messages delivered.

Figure 5.3 shows the number of Messages Delivered and the number of Interested Nodes in each topic. It must also be noted in this figure, that from day 9 to 11, there are no interested nodes, so no messages are delivered.

Additionally, in Table 5.3 it can be seen the total number of Messages Delivered and Relayed by the three protocols at the end of the simulation (day 17).

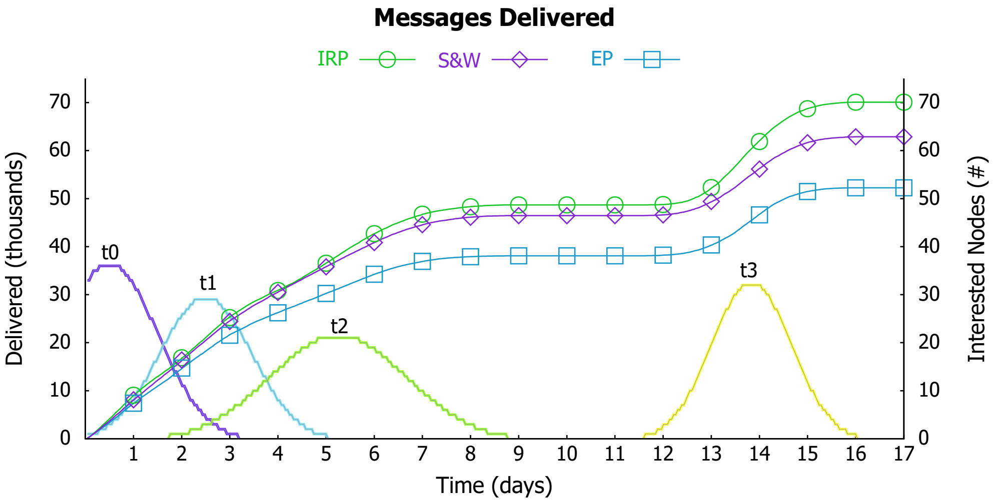


Fig. 5.3: Number of Messages Delivered by *IRP*, *S&W* and *EP* in each topic.

We observe that *EP* delivers far fewer messages than the other two algorithms, approximately 26% less than *IRP*. This is because it is forwarding messages about all topics to all nodes regardless of the nodes' interest in receiving a message, filling their buffers with as many copies of the generated messages, and dropping these messages before they reach their destinations.

The number of messages delivered by *S&W*, is approximately 11% lower than those delivered by *IRP*, because, as *S&W* also forwards messages to all nodes it encounters, this fills the buffers, causing that some of the limited number of copies of the messages to be dropped from the network and not to reach nodes that might be interested later.

	Messages Delivered			Messages Relayed		
Topic	*IRP*	*S&W*	*EP*	*IRP*	*S&W*	*EP*
t0	14,451	12,772	11,437	20,200	47,146	119,023
t1	15,797	15,687	13,088	21,071	49,873	120,954
t2	18,455	18,010	13,567	23,963	51,654	121,409
t3	21,386	16,417	14,170	26,640	50,609	120,669
t0-t3	70,090	62,884	52,263	91,876	199,281	482,055

Tab. 5.3: Total number of Messages Delivered and Relayed by *IRP*, *S&W* and *EP* in each topic.

As we have just seen, *IRP* obtains the best results, delivering the most messages about each topic. It is because it is using its knowledge of the interest and its perception of the topics to make forwarding decisions and deliver messages, not wasting efforts in forwarding messages to every node it encounters. It is important to highlight that from day 9 to 11, since there is no interest in any topic, when using *IRP* no messages are relayed as it will be seen next.

5.2.2 Messages Relayed

The Messages Relayed metric gives us a measure of the efficiency of *IRP* for reducing network overhead compared with the other protocols.

Figure 5.4 shows the number of Messages Relayed by the three protocols and the number of Interested Nodes in each topic.

The messages relayed by *EP* are thousands (not visible in the figure 5.4), approximately 424.7% more messages than *IRP*, because it relays them indiscriminately to the nodes, and it quickly fills the buffers and discards the messages regardless if the messages are of interest to the nodes or not, but the already existing copies are relayed again.

The number of messages relayed by *S&W* grows linearly and reach up to 116.9% more messages than *IRP*, this happens because it relays messages to all nodes without considering their interest, but limited by a maximum number of message copies per message.

Unlike *EP* and *S&W*, *IRP* relays the least amount of messages, forwarding more messages only when and where there is interest or high perception of interest in a topic (higher than the threshold), and not forwarding messages when there is no interest and low perception, as can be seen from day 9 to 11.

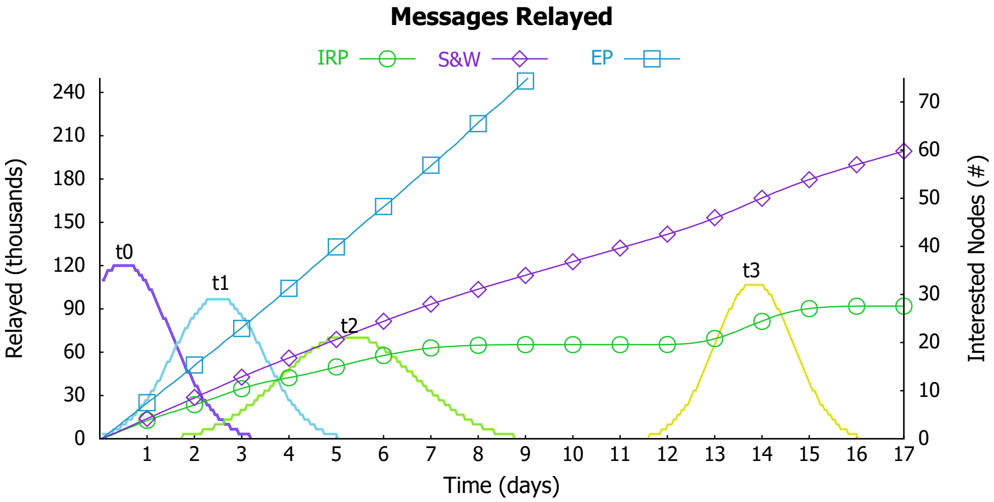


Fig. 5.4: Number of Messages Relayed by *IRP*, *S&W* and *EP* in each topic.

Additionally, also in contrast to *EP* and *S&W*, which have around the same number of Messages Relayed for each topic, it can be seen that for *IRP*, the number of Messages Relayed (and also delivered) on $t3$ is higher than in the other topics, because there are more messages on this topic in the network than there were in the other topics, since *IRP* has not deleted them yet thanks to its buffer management.

5.2.3 Buffer Occupancy

Next, we analyze the Buffer Occupancy to see how this resource is managed by the different algorithms.

Concerning *EP*, we see in Figure 5.5 that when there is no interest, the buffer occupancy for each one of the four topics is around 20 to 25% (1/4). It can also be seen that the buffer occupancy on a topic grows up to 40% when there are interested nodes. This is because in *EP*; when there are interested nodes, the messages received by those nodes are put back in its buffer just one more time to try to deliver them. For this reason, in *EP*, the buffer increases slightly when there is interest in a topic. Afterward, since the older messages are removed to receive the newer ones, the buffer occupancy for that topic returns to 20-25%.

The Buffer Occupancy in *S&W* is shown in Figure 5.6. Its behavior is similar to that of the buffer in *EP*, because in order to receive new messages both are removing messages regardless of the interest of the nodes. The buffer is filled up to 40% for each topic when there are interested nodes, but then it drops and remains at approximately the same percentage between 20 and 25%, just as *EP* does during the whole simulation.

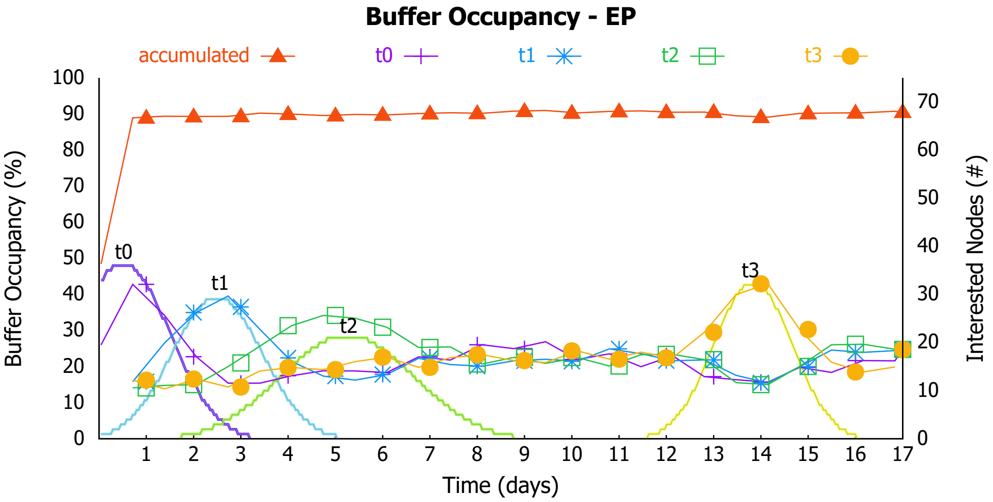


Fig. 5.5: Percentage of Buffer Occupancy of *EP* in each topic.

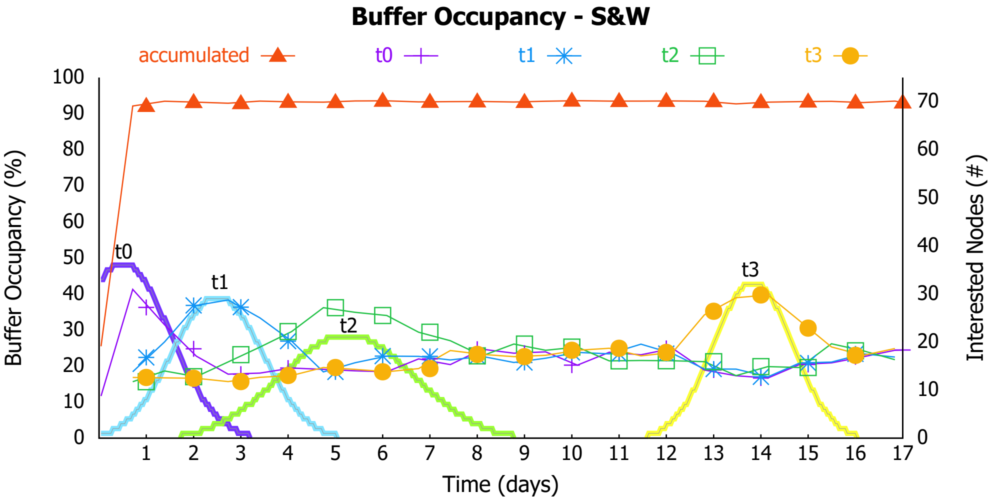


Fig. 5.6: Percentage of Buffer Occupancy of *S&W* in each topic.

Now, we analyze *IRP*'s Buffer Occupancy. Figure 5.7 shows the percentage of the buffer that is occupied by the messages of each one of the topics. It can be noted that unlike *S&W* and *EP*, while nodes become and remain interested, the buffer fills up with the messages of interest and its occupancy grows to 90%, and then practically all those messages are removed when another topic is of interest, following approximately the same behavior as the interest of the nodes.

To see how the occupancy of the buffers evolves, we can analyze $t2$. First, while the nodes are interested in a topic, this topic's occupancy of the buffers grows. After the interest in it ends, and if there is no interest in any other topic, the buffer continues full of messages of the topic. This is partly because the messages' lifespan is associated with the interest that exist in the network (messages have infinite TTL), and they are not deleted for being too old. This is also due to the fact that since

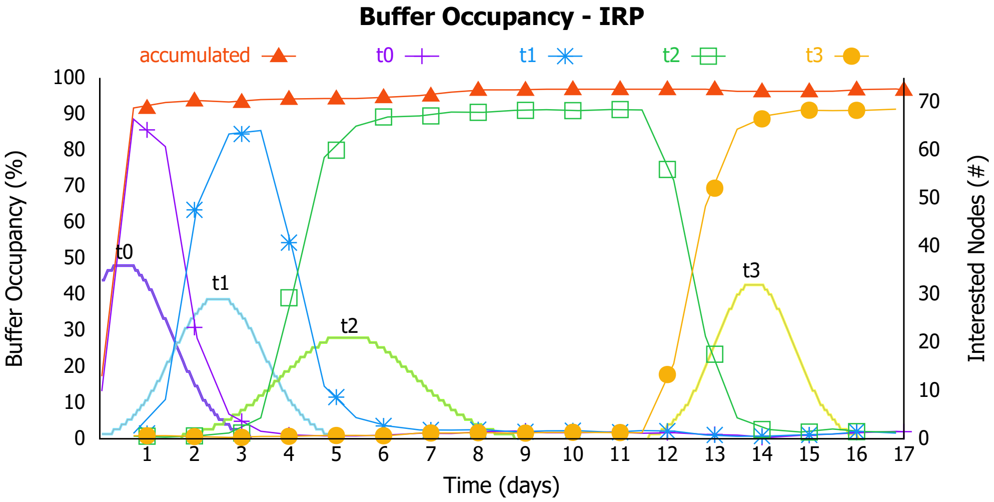


Fig. 5.7: Percentage of Buffer Occupancy of *IRP* in each topic.

there are no interested nodes in other topics yet, *IRP* does not forward messages and there is no need to drop messages.

When the interest in another topic starts, for example, $t3$ at day 12, the buffer occupancy of messages in topics that are no longer interesting decreases, while it increases in the interesting one.

We can see that the behavior of the buffers in all the other topics is the same as in $t2$, being the buffers filled mainly with the messages of the topics in which there is interest in the network at each moment.

As a summary, we can see that in *IRP*, each time there is interest in a topic, the Buffer Occupancy in it increases, and it decreases when interest in other topics appears. This way, the buffer is always filled with messages of the topics that most of the nodes are interested in. Instead, both *S&W* and *EP* maintain most part of their buffers filled with non-interesting messages, which reduces their efficiency.

5.2.4 Messages Latency

The last metric we analyze is the Message Latency obtained by the three algorithms, that is shown in Figure 5.8 together with the number of Interested Nodes.

EP has a latency of about 5 hours. This is due to the fact that although it delivers the least messages, some of them have been circulating in the network for a long time, given that *EP* does not have any restriction on the number of copies made.

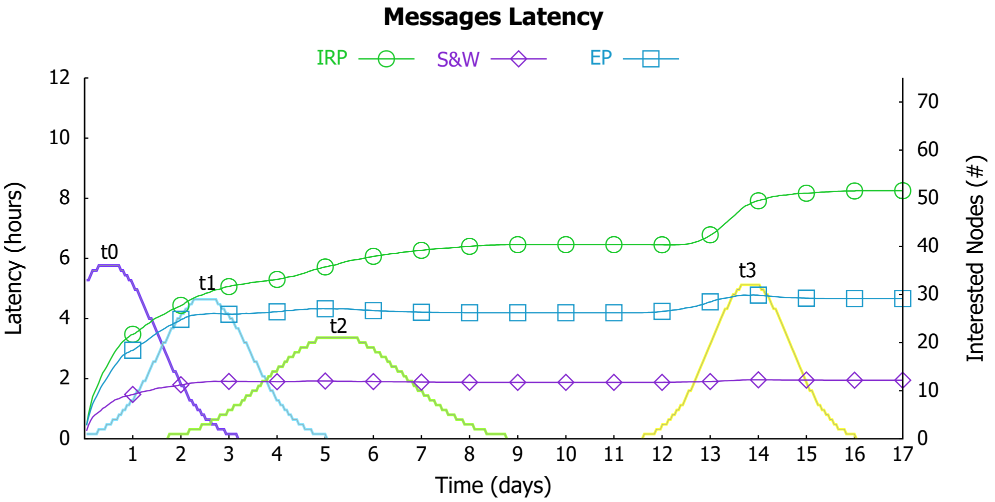


Fig. 5.8: Messages Latency obtained by *IRP*, *S&W* and *EP*.

The latency obtained by *S&W* is approximately of 2 hours, which is the lowest, from day 1 and onward. This is explained by the fact that it quickly fills the buffers, and therefore it has to discard the older messages, so it mainly only delivers the most recent messages generated.

Concerning *IRP*, we observe that its latency increases with time, reaching a maximum of approximately 8 hours at the end of the simulation. Although the latency of *IRP* is the highest, this only confirms its desired behavior. As it does not overflow the buffers with messages of no interest, there is no need to remove them from the network. Then, when there is interest in them, they can be forwarded and delivered, even those created few hours before the interest in each topic begins (unlike *S&W* and *EP*, who simply drop and never deliver them), making the latency to increase. It must be noted that with this behavior, with *IRP* users receive messages that the other protocols have already dropped.

5.3 Summary of the results

In this chapter, we presented the experiments and results in order to evaluate the performance of *IRP*. We compared the obtained results about four different topics with *S&W* and *EP*.

In terms of Messages Delivered, *IRP* outperformed the *S&W* and *EP* protocols, delivering the 11% more messages than *S&W* and 26% more messages than *EP*. About Messages Relayed, *IRP* proved to be the most efficient, as in total it forwarded the 116% less than *S&W* and 424% less than *EP*.

Regarding *IRP* Buffer Occupancy, it changed depending on the interest of the nodes. When nodes were interested in a topic, the buffer filled up with messages of interest, reaching up to 90% occupancy. This behavior was in contrast to *S&W* and *EP*, which maintained relatively constant buffer occupancy levels, ranged from 20% to 25% for each topic. This was because both protocols removed messages indiscriminately, regardless of interest.

We also saw that *IRP* Latency increased with time, reaching a maximum of approximately 8 hours at the end of the simulation. *S&W* achieved only a quarter of the latency of *IRP* and *EP* more than half. This was due to *IRP*'s strategy of not overloading the buffers with uninteresting messages, which results in the delivery of older messages when the interest in some topic began to increase. However, the other two, filled their buffers quickly and mainly delivered the most recent messages, discarding the oldest ones.

In order to summarize, once analyzed and compared the performance of *IRP* with the other two protocols, *IRP* accomplished its purpose of delivering more messages using its knowledge about user's interest and relayed fewer messages than the other traditional protocols. This was because *IRP* was designed to relay, and so deliver, more messages when and where there is interest in a topic, not spending effort relaying messages if there is no interest.

6

Evaluating *IRP* and *SCORP* by using real contact traces.

A common practice to evaluate the performance of algorithms in OppNets is to compare them with traditional protocols like *EP* and *S&W* or with other more specific ones and, it is also common to do it in scenarios that use synthetic traces.

Additionally, analyzing the performance with at least one other algorithm that tries to solve a similar problem to the one designed and, perform it in a scenario that use real contact traces improves the algorithm's comparative. Therefore, in this chapter, we present the new performance evaluation in which we compared *IRP* with an interest-based algorithm in a scenario that uses real contact traces. In order to do that, we analyzed the performance of *IRP* against Social-aware Content-based Opportunistic Routing Protocol (*SCORP*) and *EP*. We explain how *IRP* operates in a scenario that uses real contact traces, and how is the interaction of nodes that have interest in the same topic in two different periods of time.

These results were presented in a Conference paper titled "Interest-based Routing in Opportunistic Networks: Evaluating IRP against SCORP" at the 19th International Conference on Wireless and Mobile Computing, Networking and Communications in 2023 [9].

6.1 Experiments

In this section, we present the experiments that we have conducted in order to evaluate IRP's performance against *SCORP*. We performed a set of experiments using *The ONE* simulator [20] and ran five different simulations using different random seeds to select interested nodes. Then, the average results were calculated.

6.1.1 Contact Model

For these experiments, it has been used the traces collected by [12] between real interaction of students in the MIT Media Laboratory, and MIT Sloan business school located in Cambridge, Massachusetts, United States. The traces contain data that represents over 350,000 hours of human behavior, with the users moving through their environment naturally while the information was being collected.

The information includes call logs, Bluetooth devices in proximity, cell tower IDs, application usage, and phone status (such as charging and idle). It contains data about communication, proximity, location, and activity, but we only used the traces of contacts and contact duration from the 100 subjects in the course of the 2004-2005 academic year.

We use these traces in order to use real interactions with persons in the University that have different interests in topics. Therefore, we can evaluate the algorithms in a real scenario and verify their performance as realistically as possible.

6.1.2 Interest model

We modelled the number of interested nodes in every topic using a trapezoid distribution [11]. We defined four topics ($t0$ to $t3$) along 16 days that start at different hours of the day, as can be seen in Figure 6.1. We split the interest in topic $t2$ into two different time periods, the first one starts at 12 a.m. and, the second one at 7 a.m. Topic t0 starts at 7 a.m. and topics t1 and t3 start at 12 p.m. Everyone reaches a different maximum, between 35% and 50% of the total number of nodes in the experiments.

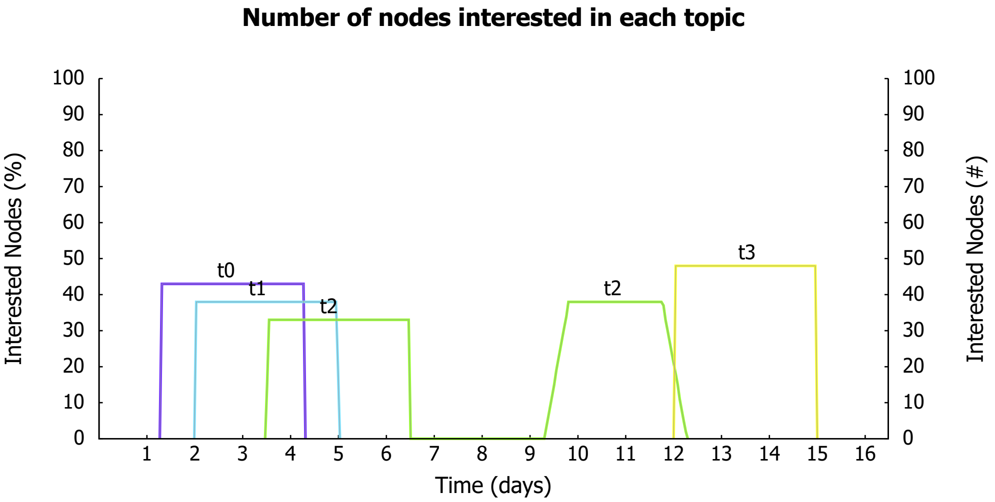


Fig. 6.1: Time distribution of the number of interested nodes for every topic: Four topics (t0 to t3) were distributed over sixteen simulation days.

The trapezoids have a different time duration. We defined topics whose interest starts at a different time but overlaps for a while (t0, t1 and t2). We also include a small-time interval (days 7-9) in which no node is interested in any topic. We split the interest in topic t2 into two different time periods, so the behavior of *IRP* and *SCORP* when an interesting topic ceases to be so but later becomes interesting again can be analyzed. From day 1 to 6, there is a high interest in more than one topic simultaneously.

The maximum number of interested nodes in a topic and the duration of the interest (the time elapsed since the first node becomes interested until there are no more interested nodes) are shown in Table 6.1.

6.1.3 Experimental Settings

The parameters shown in Table 6.1 have been used to configure the set of simulations.

	Parameter	**Value**	**Units**
Scenario	simulation time	16	days
	number of simul.	5	simul.
Network	transmission range	10	m
	speed transmission	2	Mbps
Nodes	number of nodes	100	nodes
	speed range	3 – 50	km/h
	buffer size	20	MB
Messages	size	500	kB
	speed of creation	30-36	msg/h
	TTL	∞	-
Topics	t0	72	hours
	t1	74	hours
	t2	147	hours
	t3	72	hours
IRP	α	0.3	-
	β	0.3	-
	T_f	0.8	-

Tab. 6.1: Simulation parameters.

The values of α, β and T_f can be adjusted according to the needs. We tested different values and selected those that provided the best results. We set an infinite TTL to make the message lifespan depend on the topic interest in the network.

SCORP already supports destinations based on users' interests. For this reason, no additional changes to the protocol have been required, and the published implementation for the *ONE* simulator [1] has been used.

---

[1] https://www.netlab.tkk.fi/tutkimus/dtn/theone/

6.2 Results

In this section, we present and analyze the average results obtained by five executions of the simulations in the experiments, comparing Interest-based Routing Protocol (*IRP*), Social-aware Content-based Opportunistic Routing Protocol (*SCORP*) and also Epidemic (*EP*). We analyze the Messages Delivered, Messages Relayed, Buffer Occupancy, Messages Latency and Overhead ratio.

6.2.1 Messages Delivered

We first analyze the total number of messages delivered. Figure 6.2 shows the number of Messages Delivered and the number of Interested Nodes. Additionally, Table 6.2 shows the total number of Messages Delivered and Relayed by the protocols at the end of the simulation (day 16).

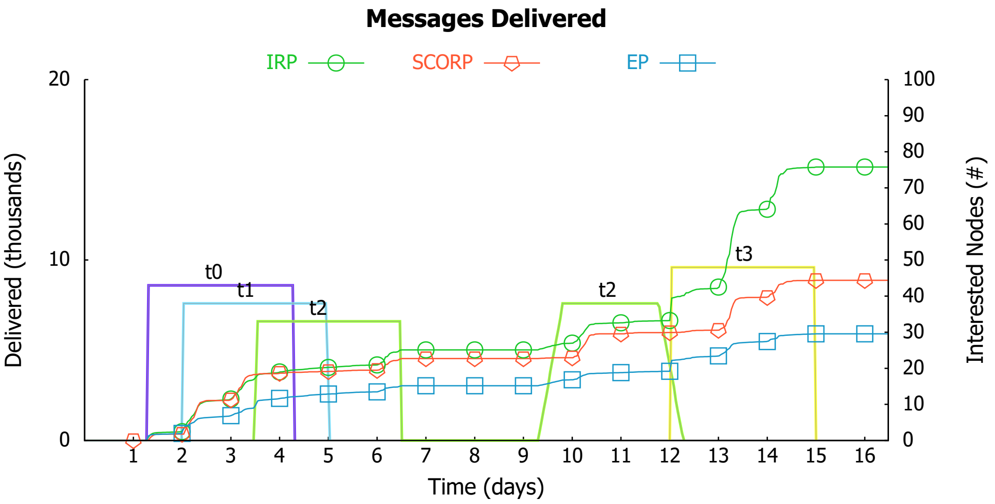


Fig. 6.2: Number of Messages Delivered by *IRP*, *SCORP* and *EP*.

We observe that *EP* delivers approximately 38% of the messages delivered by *IRP*. This happens because it forwards messages to all nodes indiscriminately, filling their buffers and making them drop messages before they reach their destinations. About *SCORP*, it delivers approximately 42% less than *IRP*. This happens because *SCORP* always tries to forward messages through the best socially connected nodes (nodes with the highest connection time with nodes of similar interest). Thus, those nodes fill their buffers and drop many messages before they can deliver them to their destinations.

At the end, *IRP* obtains the best results in delivering messages because it uses its knowledge about interest and perception of interest in the network to make forward-

	Messages Delivered			Messages Relayed		
Topic	IRP	SCORP	EP	IRP	SCORP	EP
t0	2,431	2,392	1,106	26,392	102,560	99,590
t1	1,224	1,395	941	2,364	104,665	103,645
t2	3,003	2,201	1,798	3,408	91,908	91,832
t3	8,495	2,885	2,064	13,505	55,021	123,902
t0-t3	15,153	8,873	5,909	45,669	354,153	418,969

Tab. 6.2: Total number of Messages Delivered and Relayed by *IRP*, *SCORP* and *EP* in each topic

ing decisions and deliver messages, not wasting efforts in forwarding messages to every node it encounters.

6.2.2 Messages Relayed

Next, we analyze the number of Messages Relayed, which gives us a measure of how many messages the algorithms generate in order to deliver the messages.

Figure 6.3 shows the number of Messages Relayed by the algorithms and the number of Interested Nodes in each topic.

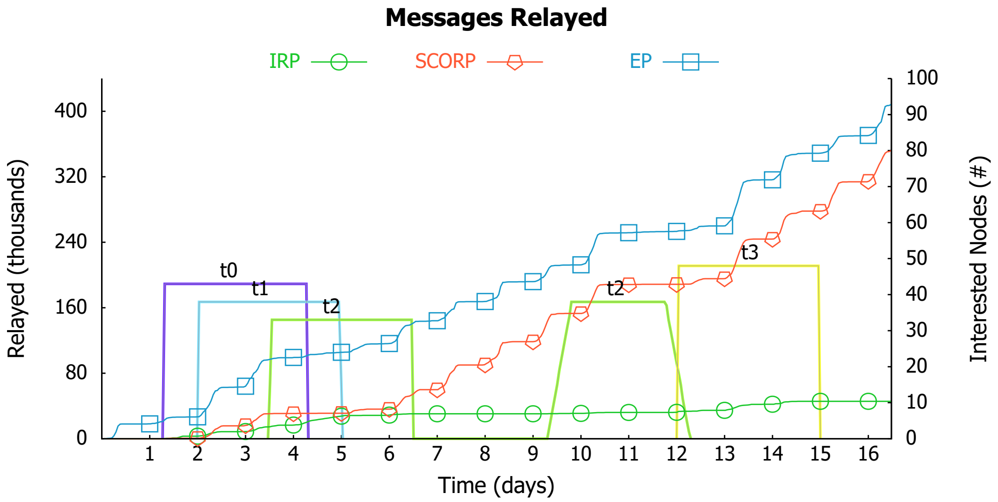


Fig. 6.3: Number of Messages Relayed by *IRP*, *SCORP* and *EP*.

We can see that *IRP* and *SCORP* forward messages only after nodes become interested. *EP* relays messages from the beginning and to all encountered nodes regarding their interest in a topic, so the number of messages it relays is about 9 times higher than that relayed by *IRP*, and similar to that of *SCORP*. The number of messages relayed by *SCORP* is also higher than that of *IRP*. This happens because, although *SCORP* is good at detecting that an interest on a topic starts or exists, it does not have a mechanism to detect when to stop forwarding messages because there is no more interest, as *IRP* does. We see this behavior between days 7 and 9, when *SCORP* continues to forward messages, and *IRP* stops doing so.

On the other hand, *IRP* relays the least amount of messages (12% and 10% regarding *SCORP* and *IRP* respectively), forwarding more of them only when and where there is interest or high perception of interest in a topic (higher than the threshold), and not forwarding messages when there is no interest and low perception.

6.2.3 Buffer Occupancy

Next, we analyze the Buffer Occupancy to see how the different algorithms manage this resource.

Figure 6.4 illustrates the daily level of buffer occupancy. It must be noticed that all-time *EP* fills its buffers with approximately 25% of the messages in each topic regardless of the interest in the network, and when it is full, it starts dropping old messages in order to receive the new ones. Therefore, it loses messages that might be of interest later on. Regarding *SCORP*, it starts to fill its buffer with the messages of the topics that appear. As *SCORP* does not identify which topics are currently interesting than others, it also fills it with messages from all topics all time, dropping the old ones to receive the new ones.

On the other hand, while nodes become and remain interested in a topic, *IRP* fills up the buffer with messages about that topic. Then those messages are removed when there is no more interest in them, and another topic becomes interesting to the network.

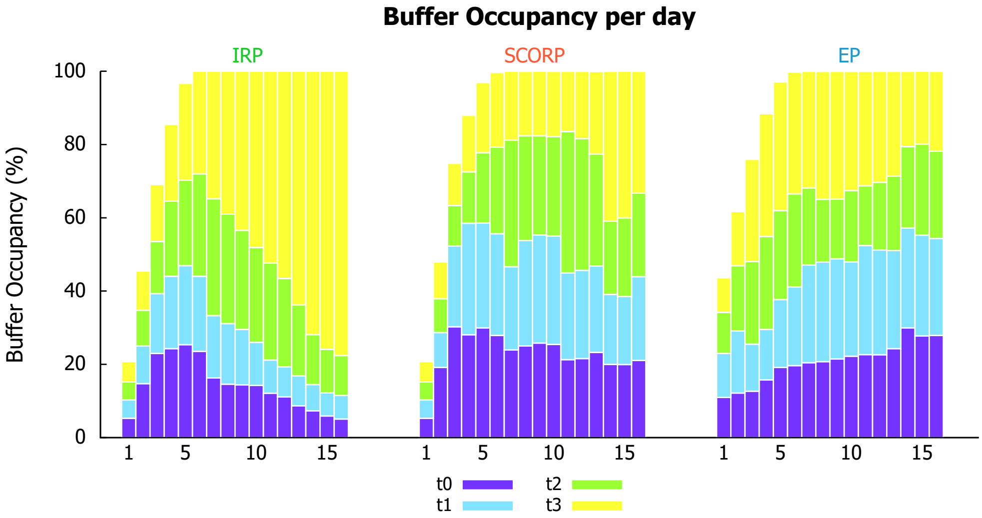


Fig. 6.4: Buffer Occupancy, by message's topic, of *IRP*, *SCORP* and *EP*.

In order to see clearly how *IRP*'s occupancy of the buffers evolves, we can analyze t2 and t3. First, while the nodes are interested in t2 (between days 4 and 12), the occupancy of the buffers by this topic increases. Then, when it ends and t3 starts to become interesting on the day 12, the nodes begin dropping messages on topics like t2, which have no more interested nodes, and fill with messages on topic t3.

Unlike *IRP*, *SCORP* does not drop messages considering the interest in the network and does not stop forwarding them when they lose the interest of the nodes. So, from day 12 onwards, it drops the older ones and does not detect that t3 is the only interesting topic. This behavior causes its buffers to continue to be filled with approximately 25% of each topic and causes it to lose a lot of messages in t3, as we will see in Figure 6.6.

As a summary, we can see that in *IRP*, each time there is interest in a topic, the Buffer Occupancy in it increases, and it decreases when interest in other topics appears. This way, the buffer is always filled with messages on the topics that most of the nodes are interested in.

6.2.4 Messages Latency

Following, we analyze the Message Latency obtained by the algorithms, shown in Figure 6.5.

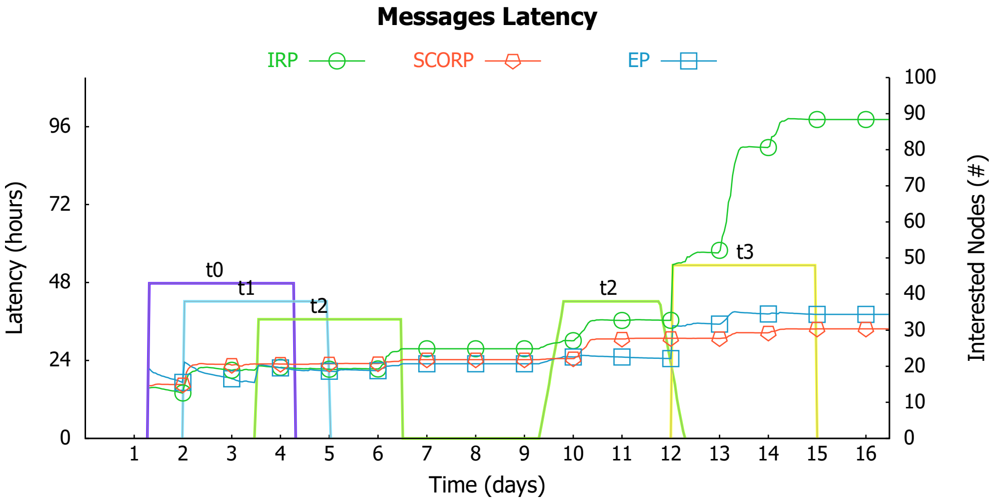


Fig. 6.5: Messages Latency obtained by *IRP*, *SCORP* and EP.

We can see in Figure 6.5 that, until day 9, all the algorithms have a similar latency. From that day on, the latency of *IRP* increases, obtaining at the end approximately three times the result obtained by *EP* and *SCORP*. This happens because, as it does not overflow the buffers with messages of no interest, there is no need to remove them from the network. Then, when there is interest in them, they can be forwarded and delivered, even those created days before the interest in each topic begins.

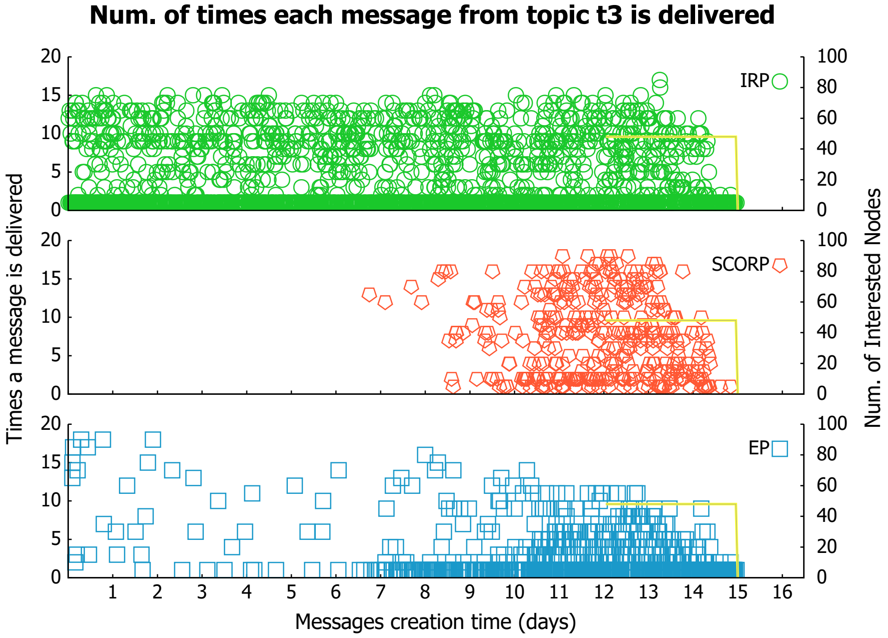


Fig. 6.6: Number of times that t3's messages have been delivered. The amount of interested nodes in every topic is also plotted. Note that messages that have not been delivered are not drawn.

EP and *SCORP* have the lowest latency, but not because they are faster at delivering messages, but because they simply drop and never deliver many of the messages created the days before the interest begins, as in the case of messages on t3.

Figure 6.6 shows how many times a single message on t3 is delivered and when it was created. We see that *IRP* delivers many of them regardless of the day they were generated, so, with *IRP*, users receive messages that *SCORP* and also *EP* have already dropped.

6.2.5 Network Overhead

Finally, we analyze the network overhead caused by each algorithm as a relation between messages relayed and messages delivered. In this way, we evaluate the resources consumed by an algorithm to deliver a message.

Figure 6.7 shows that *EP* has the worst results because it relays messages to every node it encounters regardless of their interest, collapsing the network with messages that nodes do not want to receive. *SCORP* also has a high overhead because it never stops forwarding messages even if no node is interested in them, as in the case of topics t0 or t1 from day 6 onwards. However, even if until day 6, both *IRP* and

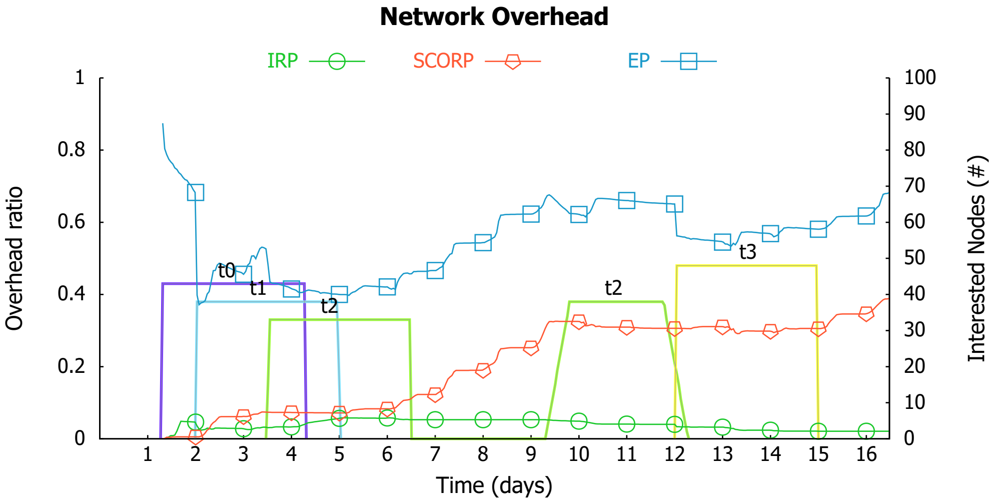


Fig. 6.7: Network Overhead obtained by *IRP*, *SCORP* and *EP*.

SCORP have a quite similar overhead, from that day on, *IRP* has the best one. This happens because it only forwards messages when it perceives that there is high interest in the network about a topic and stops forwarding when it does not, unlike *SCORP* which continues to forward messages. This way, *IRP* saves resources while few or no nodes are interested in a topic.

6.3 Summary of the results

We presented, in this chapter, the performance evaluation of *IRP* against Social-aware Content-based Opportunistic Routing Protocol (*SCORP*) and Epidemic (*EP*) in a scenario with real contact traces.

The analysis of the results indicated that *IRP* outperformed *EP* by delivering 38% more messages and *SCORP* by delivering 42% more messages. This fact was due to *IRP*'s ability to make forwarding decisions based on users interest.

IRP relayed the fewest messages, approximately 12% and 10% in comparison to *SCORP* and *EP*, respectively. We observed that *IRP* and *SCORP* relayed messages only after nodes became interest in a topic, in contrast to *EP* that, relayed messages from the beginning.

We saw that *IRP* obtained higher latency, while *EP* and *SCORP* showed lower. This was due to the fact that these two algorithms discard many messages and thus deliver practically the most recent ones.

IRP proved to have the most efficient operation, because, it saved resources during periods when few or no nodes expressed interest in particular topics, while *SCORP* continued to forward messages indiscriminately.

Contrasting *IRP* with *SCORP* allowed us to evaluate the different approaches of these protocols, especially highlighting *IRP*'s ability to detect when to stop forwarding messages, an aspect that is not present in *SCORP*.

In order to summarize, after analyzing the performance of *IRP* in comparison with *SCORP* and *EP* by using a scenario with real contact traces, *IRP* achieved its objective of delivering more messages while relaying fewer. It optimized its resources by not wasting effort by relaying messages when the interest decreased.

7

Using the redefined Metrics to evaluate *IRP*

TRADITIONALLY, in order to evaluate the performance of algorithms, metrics like Delivery Ratio and Latency are used. But, when message destinations change according to the users' interest in a topic and there are periods in which nodes are not interested in receiving messages, these metrics, calculated in the traditional way, do not provide good enough information about algorithm performance. Therefore, in this chapter we present a new performance evaluation of *IRP*, using the Delivery Ratio and Latency redefined in Chapter 4, which takes into account these considerations.

In order to make the analysis, we designed a scenario that simulates a typical daily work routine in the Eixample district and its surroundings in Barcelona. In it, there were included the interaction between users who were interested and disinterested in eleven different topics with high and low levels of interest.

Then, we calculated the Delivery Ratio and Latency by using the redefined metrics for *IRP*, *SCORP*, *S&W* and *EP*, and analyzed and compared the obtained results. In addition, a comparative in terms of Messages Delivery, Messages Relayed, Buffer Occupancy and Efficiency is also presented.

Finally, based on the results obtained, we wrote two articles that were submitted to two different JCR Journals, and actually they are under review state. The first one titled "Interest-based Routing (IRP): An efficient routing strategy for receiving messages based on users' interests in OppNets" was submitted to IEEE Transactions on Mobile Computing Journal. And, the second one titled "Redefining Metrics for an Efficient Interest-based Routing in Opportunistic Networks" was submitted to Computer Networks Journal.

7.1 Experiments

In this section, we present the experiments that we have conducted in order to evaluate *IRP*'s performance using the new proposed metrics. First, we describe the map we used and the nodes' mobility model. Next, we explain how we have modelled the interest of the users in every topic. Finally, we present and discuss the configuration settings.

7.1.1 Map

Our experiments use the actual map of the Eixample district of Barcelona (Spain) and its surroundings. The map used (see Figure 7.1) is 4.2 km wide and 2.3 km high, with an effective area of 9.66 km^2. We selected several uniformly distributed locations in the map, where we placed homes, markets and workplaces, where users interact on a daily routine.

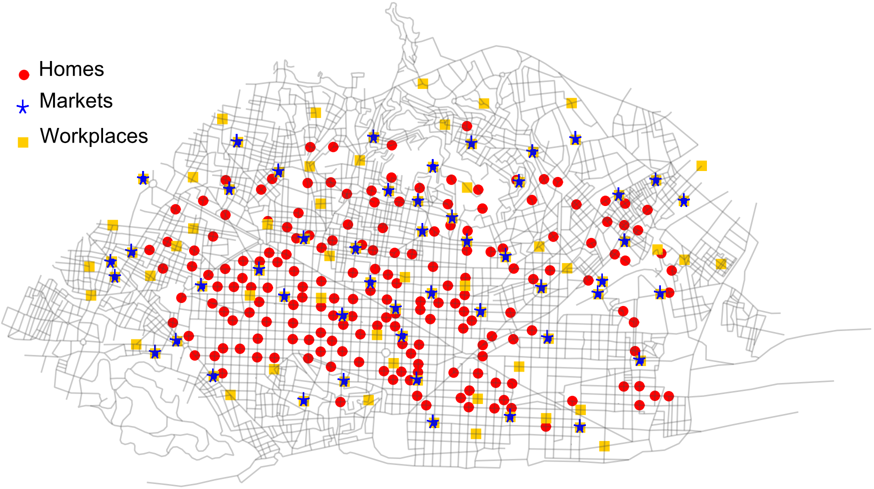


Fig. 7.1: Map of the Eixample district and its surroundings in Barcelona with the homes, markets, and workplaces set for our experiments.

7.1.2 Mobility model

We define a synthetic mobility model that aims to reflect the people's daily routine on the real map of Eixample in Barcelona. The five different roles that we have defined, and their characteristics, are explained below:

1. Office workers, who commute every day to their workplace in the morning, perform their tasks and return to their homes 8 hours after arriving at the workplace. They start activities from 8 a.m. to 10 a.m.

2. Market workers, who work at market locations. They start their 8-hour duty in different time slots from 8 a.m. to 12 a.m.

3. Market suppliers, who form part of the supply chain of the markets. They work for 10 hours, visiting the same market locations every day, before returning home.

4. Home delivery drivers, who deliver packages to homes (e.g., Amazon delivery workers). They also work for 10 hours, but they visit different homes every day.
5. Teleworkers and people at home (e.g., self-employed, unemployed, or elderly) who stay mostly at home during the day. They go to markets near their home or to take a walk for a maximum of three hours at any time between 7 a.m. and 18 a.m.

The movement model of Office workers, Market workers, Teleworkers and People at home were simulated using the Working Day Mobility model [13] implemented in *The ONE* simulator [20], while the movement model of Market suppliers and Home delivery drivers were simulated using a new class named Delivery Movement that we implemented based on the Map Based Movement in the *The ONE*. Note that during the night hours, all nodes are in their homes and there is no movement, so the only contacts are between those living in the same home.

7.1.3 Interest model

The interest of the entire group of nodes in every topic also has to be modelled. For the sake of simplicity, we model it using trapezoids.

Our trapezoidal model assumes that no node is interested in a topic at the beginning of the simulation. At some point, some nodes start to become interested in it until a maximum number of interested nodes, different for each topic, is reached. Then, all of them remain interested for a while, and finally, they start losing interest until there are no nodes interested.

In our simulation, we have defined eleven different topics. Figure 7.2 shows the time distribution and the maximum number of interested nodes for each one of them. Every time the simulator needs to select a node to be interested or disinterested in a topic in order to fit the topic's trapezoid, a node is selected randomly.

The five topics of higher interest (note that topic t1 grows twice) reach a peak between 15%, and 35% of the total number of nodes (1,000), and the six less interesting topics do not exceed 1%.

In order to try to cover all different cases, we have interest periods of time of different duration and starting at different hours of the day. We defined two topics whose interest start simultaneously at 7 a.m. (t0 and t1), topics whose interest starts at a different time at 12 a.m., 5 p.m. and 8 p.m. respectively, but overlap for a while

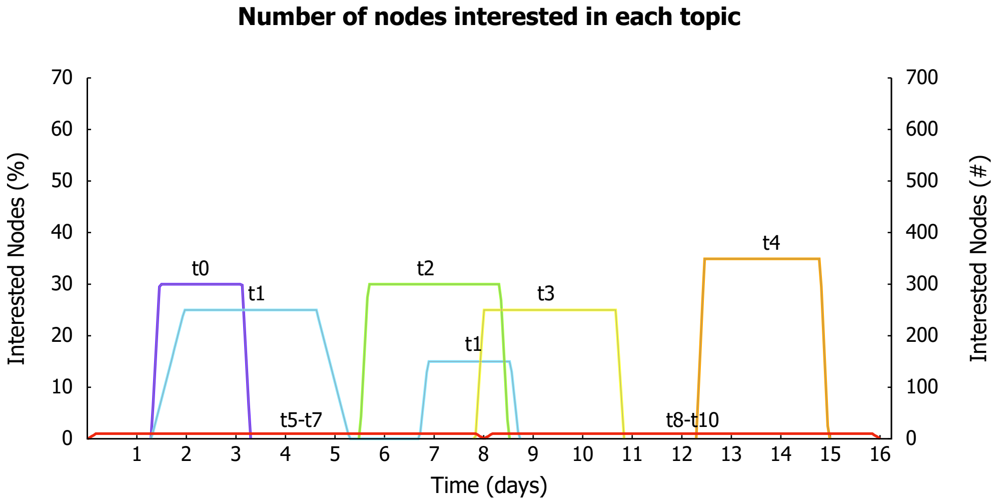


Fig. 7.2: Time distribution of the number of interested nodes for every topic: Eleven topics (t0 to t10) were distributed over sixteen simulation days, covering a wide spectrum of different situations.

(t2, t1 and t3), and a topic whose interest does not coincide in time with any other (t4), which also starts at 7 a.m.. We also included two time gaps, a small one (days 5-6) and a wider one (days 11-12), where very few or no nodes are interested in any topic. Besides, we split the interest in topic t1 into two different time periods, so the behavior of *IRP* and the other algorithms when an interesting topic ceases to be so but later becomes interesting again can be analyzed.

Also, for analyzing the behavior of the algorithms regarding the topics of less interest, we grouped them into two groups of eight days (topics t5 to t7 until day 8, and t8 to t10 from that day on).

Finally, in order to facilitate a clearer analysis of the results in Section 7.2, we named the topics that reach a high number of interested nodes (topics t0 to t4) as “interesting topics” and the topics that reach a low number of interested nodes (topics t5 to t10) as “uninteresting topics”.

7.1.4 Experimental Settings

In order to evaluate *IRP*’s performance, since none of the cited algorithms shown in Table 2.1 in Chapter 2 support all the required characteristics together, we analyzed *IRP*’s results and compared them with the ones obtained by *EP*, *S&W* and *SCORP*. Although *EP* and *S&W* are maybe two of the most basic ones, they are a good choice for the analysis since most researchers use them for performance evaluations [15].

Finally, *SCORP* is a good option because, like *IRP*, its operation is based on user's interests.

We used *The ONE* to run the simulations. We ran five different simulations using different random seeds, and the average results were calculated. Table 7.1 shows the configuration parameters used in the simulator.

	Parameter	**Value**	**Units**
Scenario	dimensions (w $*$ h)	$4.2 * 2.3$	km
	area	9.66	km^2
	simulation time	16	days
	simulations	5	simulations
Locations	homes	180	locations
	markets	50	locations
	workplaces	100	locations
Roles	office workers	320	nodes
	market workers	320	nodes
	market suppliers	80	nodes
	home delivery	80	nodes
	teleworkers	200	nodes
Nodes	number	$1,000$	nodes
	speed range	3 - 30	km/h
	transmission range	10	m
	transmission speed	2	Mbps
	buffer size	20	MB
Messages	size	400	KB/message
	creation speed	36	msg/h
	TTL	∞	-
	topics	11	topics
IRP	α	0.3	-
	β	0.3	-
	T_f	0.8	-
Spray & Wait	L	120	messages

Tab. 7.1: Simulation parameters.

As previously explained, the values of α, β and T_f can be adjusted according to the needs of each case. In our experiments, we previously ran a set of 29 simulations using various combinations of α, β and T_f to determine the one that obtained the best results.

The value of L in *S&W* was selected according to the authors' suggestion [39] of using a number between 10% and 15% of all nodes used in the experiments. Additionally, we use an infinite *TTL* in order to demonstrate that the lifespan of the message depends on the interest in the topics of the messages in the network.

Also, we had to slightly adapt the behavior of Epidemic (*EP*) and Spray & Wait (*S&W*) to a scenario in which messages are not directed to nodes, but are delivered to them depending on their interest. In this sense, we made the following changes.

- When a message is forwarded to a node, it is analyzed whether its topic is one of the node's current interesting topics, and if it is, the message is considered as delivered to that node.

- When a node becomes interested in a topic, its buffer is scanned to find all messages of that topic. All these messages are considered as delivered. No messages are removed from the buffer when doing this.

- Moreover, the messages forwarded to a node are always added to its buffer, even if the message is considered as delivered. This way, all messages can be forwarded to other nodes.

With regard to *SCORP*, it already supports destinations based on users interests. For this reason, no additional changes to the protocol have been required, and the published implementation for the *ONE* simulator [1] has been used for the simulation.

7.2 Results

In this section, we present and analyze the average results obtained by five executions of the simulations in the experiments, comparing Interest-based Routing Protocol (*IRP*), Epidemic (*EP*), Spray & Wait (*S&W*) and Social-aware Content-based Opportunistic Routing Protocol (*SCORP*). For this analysis, we study the Messages Delivered, Messages Relayed, Delivery Ratio, Buffer Occupancy, Messages Latency and finally the Efficiency of each routing algorithm.

7.2.1 Messages Delivered

The first metric we analyze is the number of Messages Delivered by each of the four routing protocols. This metric allows us to check the efficacy of *IRP* in delivering messages with respect to the other algorithms analyzed. Figure 7.3 shows the number of Messages Delivered on interesting topics (t0 to t4).

We observe that *SCORP* delivers almost one-third of the messages delivered by *S&W* and *IRP*, but a similar number of messages than *EP*. This happens because *SCORP* always tries to forward messages through the best socially connected nodes (nodes with the highest connection time with nodes of similar interest). Thus, those nodes fill their buffers and drop many messages before they can deliver them to their destinations.

---

[1] https://www.netlab.tkk.fi/tutkimus/dtn/theone/

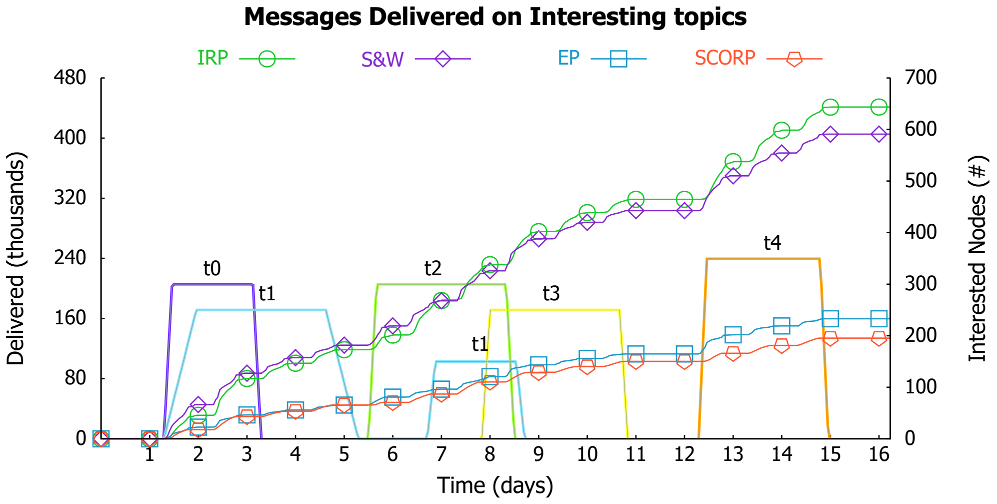


Fig. 7.3: Total number of messages delivered by *IRP*, *EP*, *S&W* and *SCORP* on interesting topics.

Messages delivered by *EP* are fewer than the messages delivered by *S&W* and *IRP* algorithms. Also, we see that it delivers a similar amount of messages than *SCORP*. This happens because it forwards messages to all nodes indiscriminately, filling their buffers and making them drop messages before they reach their destinations.

Regarding *S&W*, it delivers much more messages than *EP* and *SCORP*, although slightly fewer than *IRP*, because it limits the number of message copies it spreads over the network, so it drops fewer messages, and they can reach more interested nodes.

Finally, *IRP* delivers more messages about interesting topics than the other three algorithms (a similar amount than *S&W*, but much more than *EP*) because, as we will see later in subsection 7.2.2, it relays messages only when and where there is interest in a topic, and it manages its resources better by focusing mainly on topics that are perceived as interesting for the entire network.

In order to illustrate this, we can focus on what happens between days 11 and 12, when there are no interested nodes in any interesting topic, so no messages are delivered. Anyway, *EP* (millions of relays), *S&W* and *SCORP* (around hundred thousand relays) continue relaying messages, while *IRP* almost does not (around two thousand relays [2]), avoiding wasting resources.

Table 7.2 contains the total number of Messages Delivered and Relayed by the four routing protocols on interesting topics. These values are the ones at the end of the

---

[2] Due to the logarithmic scale, this is not easy to spot on Figure 7.5, but we have obtained the values directly from the simulation results.

simulation (day 16). We can see in Table 7.2 that *S&W* only delivers more messages than *IRP* in topic t0. This only happens at the beginning of the simulation, when *S&W* has its buffers empty, and can benefit from it by distributing copies of the all generated messages without dropping messages because the buffers are not full yet.

	Messages Delivered				Messages Relayed			
Topic	IRP	S&W	EP	SCORP	IRP	S&W	EP	SCORP
t0	50, 383	53, 282	18, 416	17, 282	80, 479	186, 548	12, 568, 385	301, 922
t1	98, 303	97, 334	33, 222	40, 503	108, 234	228, 909	8, 504, 232	528, 201
t2	96, 327	83, 699	30, 727	22, 671	126, 654	215, 238	18, 168, 894	274, 883
t3	73, 499	71, 116	30, 527	22, 462	82, 455	203, 808	17, 953, 367	231, 404
t4	122, 773	106, 883	46, 760	31, 040	180, 870	235, 922	23, 369, 691	229, 121
t0-t4	441, 285	412, 314	159, 652	133, 958	578, 692	1, 070, 425	80, 564, 569	1, 565, 531

Tab. 7.2: Messages Delivered and Messages Relayed by *IRP*, *S&W*, *EP* and *SCORP* on interesting topics.

However, we can also see that, even though *IRP* starts to forward messages delayed, it delivers in total many more messages in interesting topics (t0 to t4); 7%, 64% and 70% more than those delivered by *S&W*, *EP* and *SCORP* respectively. Later, in subsection 7.2.6, we will analyze these values again in order to evaluate the algorithms' Efficiency.

Following, we analyze what happens with messages on uninteresting topics (t5 to t10). Figure 7.4 shows the number of Messages Delivered on uninteresting topics by the four algorithms.

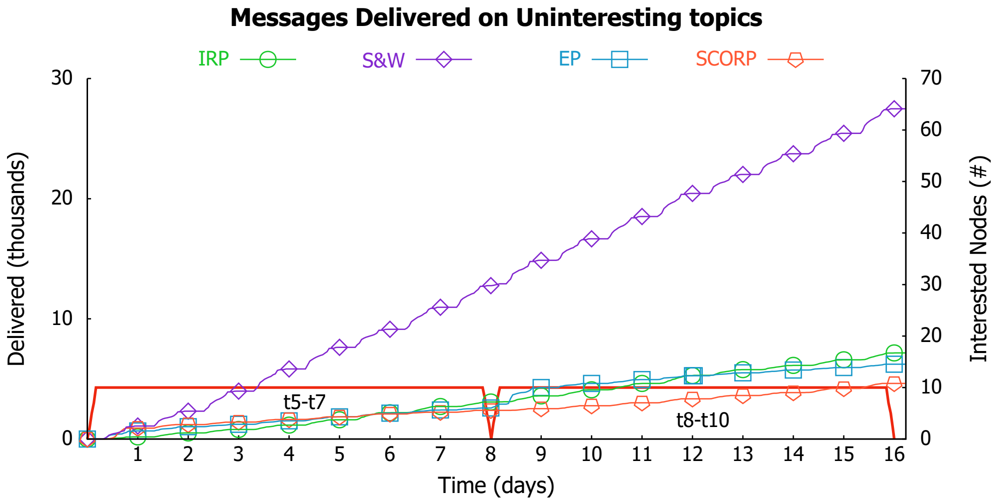


Fig. 7.4: Total number of messages delivered by *IRP*, *S&W*, *EP* and *SCORP* in topics with few interested nodes.

Again, the number of Messages Delivered by *SCORP* is the lowest of the four algorithms. Even when there is a low number of interested nodes, it continues to forward and drop messages through a few nodes that are socially better connected, filling their buffers and removing a lot of messages on interesting topics. As a result, many messages are forwarded but not delivered.

Concerning *EP*, it continues to send messages indiscriminately to all contacted nodes, so it quickly fills their buffers and repeatedly discards messages before they reach their destination.

Regarding *S&W*, it delivers the highest number of messages because it distributes only a limited number of copies of those messages over the network. Therefore, buffers are not filled as much as with *EP*, so fewer messages are discarded, and more reach the nodes interested in these topics.

When using *IRP*, nodes do not detect enough interest in the network and, by design, they only deliver messages directly to interested nodes without relaying them through intermediate nodes.

Table 7.3 shows the number of Messages Delivered and Messages Relayed by the four routing protocols on uninteresting topics. We see that *IRP* delivers only the 25% of the Messages Delivered by *S&W* in uninteresting topics (t5 to t10), but it delivers 14% more messages than those delivered by *EP* and 36% more than *SCORP*. Anyway, it must be noted that the real difference in the total number of messages delivered is very small, as there are few nodes interested in those topics.

	Messages Delivered				Messages Relayed			
Topic	IRP	S&W	EP	SCORP	IRP	S&W	EP	SCORP
t5-t7	3,083	13,739	2,580	2,383	3,083	430,214	109,041,656	1,173,323
t8-t10	4,079	14,361	3,645	2,239	4,080	432,171	98,120,911	281,701
t5-t10	7,162	28,100	6,225	4,622	7,163	862,385	207,162,567	1,455,024

Tab. 7.3: Messages Delivered and Messages Relayed by *IRP*, *S&W*, *EP* and *SCORP* on uninteresting topics.

Finally, the fact that *IRP* delivers few messages on uninteresting topics helps it devote more resources to delivering messages on interesting topics, as it focuses on forwarding messages where and when a larger number of nodes are interested in receiving them.

7.2.2 Messages Relayed

The next metric we analyze is Messages Relayed. Figure 7.5 shows the number of Messages Relayed on the interesting topics by the four studied algorithms on a logarithmic scale.

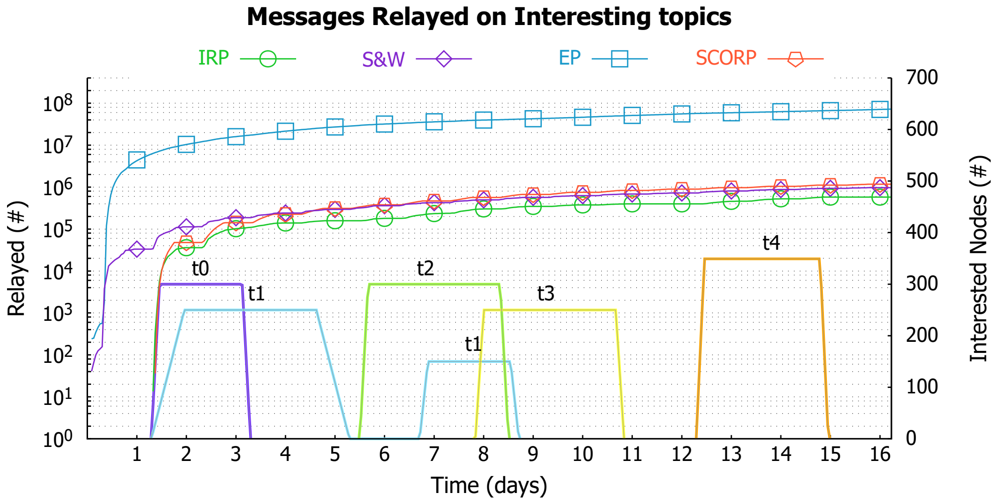


Fig. 7.5: Total number of messages relayed by *IRP*, *S&W EP* and *SCORP* on interesting topics. Note the logarithmic scale.

We can observe that both *EP* and *S&W* forward messages even before nodes become interested in a topic, while *IRP* and *SCORP* do so only after nodes become interested. Regarding *EP*, we observe that it relays millions of messages indiscriminately because it relays messages to all encountered nodes regarding their interest in a topic. 07916139J

We also see that the number of messages relayed by *SCORP* is slightly higher than *IRP* and *S&W*, but much lower than *EP*. This happens because *SCORP* does not have a mechanism to detect when to stop forwarding messages when there is no more interest. Therefore, it continues to relay messages without restriction in the number of message copies, although mainly to the nodes that it believes that are socially better connected.

The number of messages relayed by *S&W* grows linearly, and this happens because it relays messages to all nodes, only limited by the number of message copies it makes for each message.

Regarding *IRP*, it relays the fewer messages because it relays only when and where nodes perceive high interest (higher than the threshold) around it and does not forward messages when and where there is no interest. Consequently, as we have seen in the previous subsection 7.2.1, *IRP* accomplishes its purpose of delivering more messages using its knowledge about interests and globally relaying fewer messages than the other algorithms.

Next, Figure 7.6 shows the number of Messages Relayed on the uninteresting topics on a logarithmic scale. We can see that *EP* relays millions of messages indiscriminately throughout the network, compared to the other three studied algorithms. This is because it relays messages to all nodes that it encounters without any restriction.

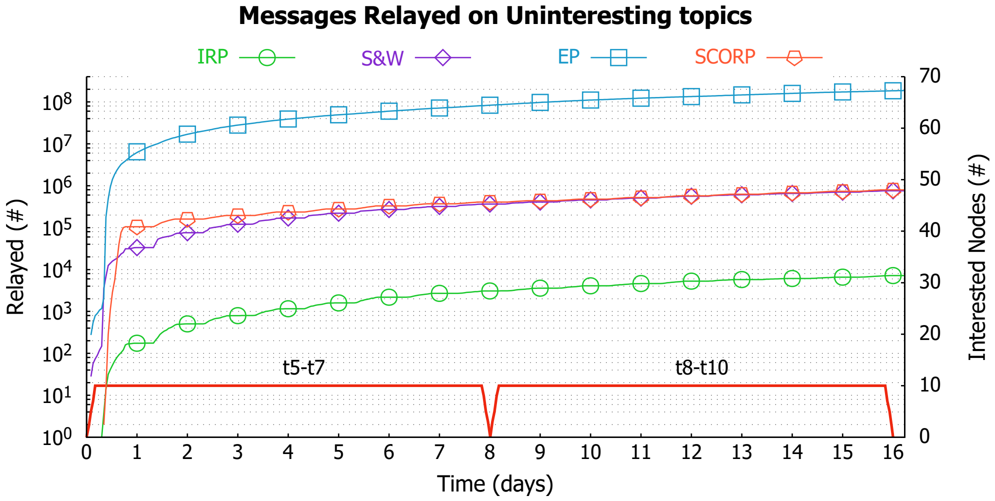


Fig. 7.6: Total number of message relayed by *IRP*, *S&W EP* and *SCORP* on uninteresting topics. Note the logarithmic scale.

We can note that *S&W* relays a similar amount of messages as *SCORP* and many more messages than *IRP*. This is because, despite knowing nothing about interests, it is limited only by the number of copies of messages that it can make.

Regarding *SCORP*, although it delivers messages using the user's interest, as we explained before, when interest ceases, it continues relaying messages to nodes with a better social connection, even if they are no longer of interest to the network.

In the case of *IRP*, it relays far fewer messages than the three other algorithms. *IRP* does not perceive enough interest and only relays messages when nodes directly encounter an interested node. Consequently, it relays far fewer messages than *EP*, *S&W* and *SCORP* and it also delivers a lower number of uninteresting messages compared with *S&W*. This behavior allows *IRP* to deliver more messages when and where there is a high interest in the network.

Finally, *IRP* saves 45.94% of message relays in interesting topics (t0 to t4) compared to *S&W*, 63.03% compared to *SCORP*, and 99.28% compared to *EP*. It relays far fewer messages in uninteresting topics (t5 to t10) than the other algorithms, saving 99.17% of relays compared to *S&W*, 99.50% with *SCORP* and 99.99% with *EP*.

7.2.3 Delivery Ratio

The next step to evaluate the performance among the four algorithms is to analyze the Delivery Ratio. As explained in Section 4.1, only the interested nodes, and only the messages generated until a node becomes disinterested, are taken into account when calculating this metric. Figure 7.7 shows the Delivery Ratio obtained by the four algorithms on interesting topics.

It can be observed that for all algorithms, and in all topics, in the early morning hours, when nodes start to move, the Delivery Ratio increases sharply, and it remains high until nodes return home in the evening. This happens because nodes begin to have contacts and therefore deliveries, and that continues during the rest of the day. Afterward, during late evening and night, the mobility decreases, resulting in fewer contacts between nodes and therefore fewer deliveries as well. However, new messages continue to be generated, leading to a gradual diminution of the Delivery Ratio until the next morning. This night trend becomes less noticeable as the number of days increase, due to the greater number of messages in the network, so that the messages generated at night represent a lower percentage of the total of messages.

In the case of *IRP*, when interest in a topic begins, it must first learn about this interest and build its perception to start forwarding messages to deliver them, which cannot happen for topics that begin to be of interest late in the day (evening and night) when there is low mobility and few contacts between nodes. Thus, it always starts forwarding and thus delivering messages later than *EP* and *S&W*, which start distributing copies of messages through the network as soon as they are created.

However, in topics t1, t2, t3, and t4, *IRP* slightly outperforms the Delivery Ratio of the other three algorithms, despite its late start. This is because the messages previously distributed by both *EP* and *S&W* have already flooded the network, collapsing the buffers of all nodes and losing some messages before they could be delivered to the interested nodes when interest starts.

Note that by not forwarding messages on uninteresting topics, *IRP*'s buffers are mostly filled with messages on interesting topics. So *IRP* can forward more messages on interesting topics, and it can deliver more of them, with better overall results and far fewer message relays than the others.

Regarding *S&W*, it gets slightly better overall results than *IRP* in topic t0 and the first period of interest in t1, while being only slightly worse in the rest of the topics. This happens because *S&W* benefits from having empty buffers at the beginning of the

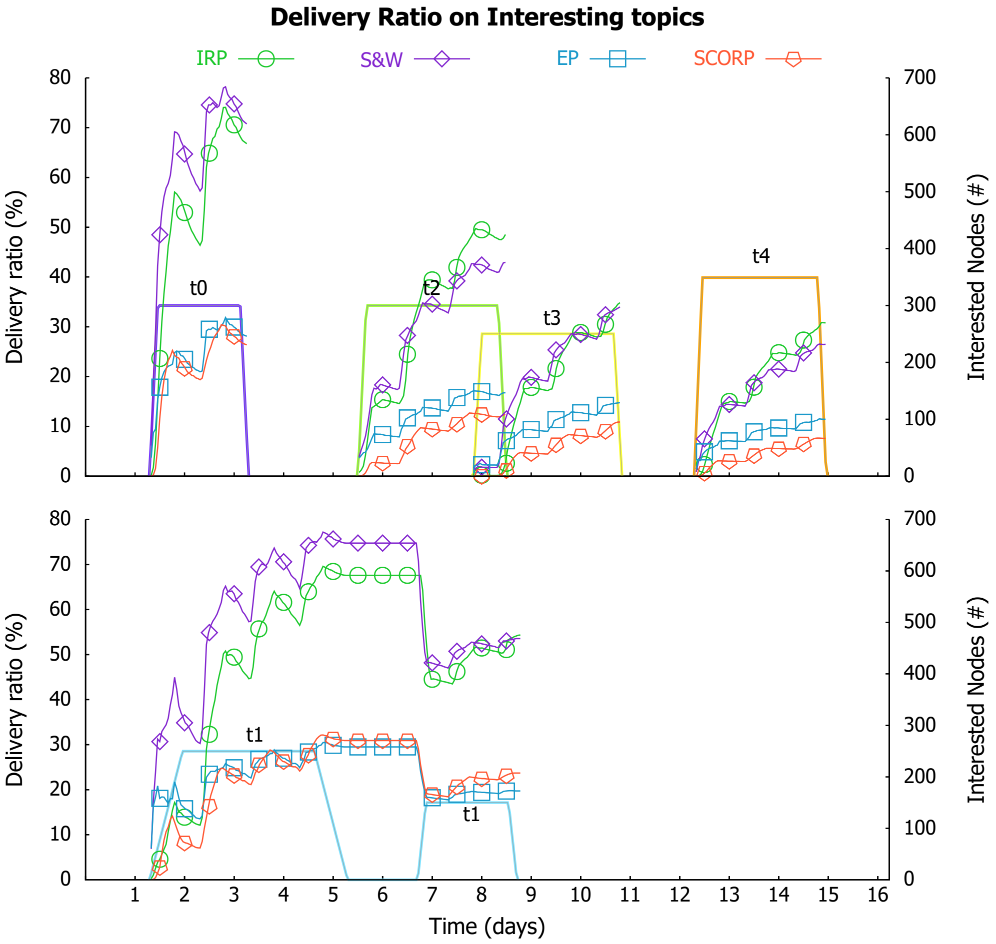


Fig. 7.7: Delivery Ratio obtained by *IRP*, *S&W*, *EP* and *SCORP* on interesting topics. In order to visualize the behavior of t1 clearly, we have plotted it separately from the other topics. Note that interest in topic t1 overlaps with interest in topics t0, at the beginning of the simulation, and t2 and t3, at the middle of the simulation

simulation, and it distributes many copies of the messages generated before nodes become interested, no matter when a topic starts to be interesting for nodes.

Concerning *EP*, it only achieves slightly better results than *SCORP*. This is because it distributes millions of copies of generated messages (as shown in Fig. 7.5 and Fig. 7.6) before nodes become interested, regardless of when a topic becomes interesting. As a result, it floods the network and drops messages, delivering few of them to interested nodes.

We can see that *SCORP* has the lowest Delivery Ratio and *EP* has slightly better results, but their results are the lowest compared to the other two algorithms (*IRP* and *S&W*). This is because the nodes try to forward messages on all topics that have been interesting through the few nodes that are socially better connected, collapsing their buffers and dropping them before they can reach their destinations.

Figure 7.8 shows the Delivery Ratio obtained by the four algorithms on uninteresting topics. Since topics from t5 to t7, and from t8 to t10, have the same period of interest and maximum number of interested nodes, we have grouped them in these two groups, being the Delivery Ratio shown the average of the three topics of each group.

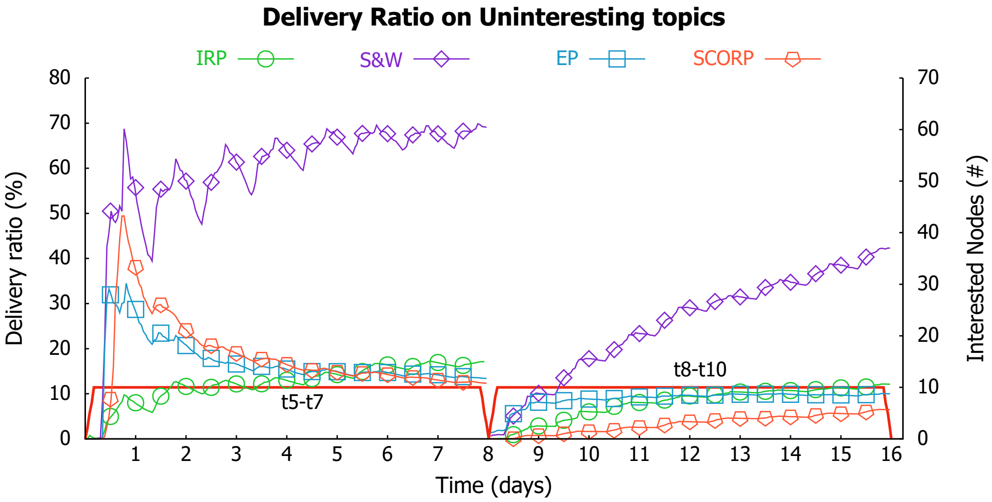


Fig. 7.8: Delivery ratio obtained by *IRP*, *EP*, *S&W* and *SCORP* on uninteresting topics. The values shown are the average Delivery Ratio of topics t5 to t7, and t8 to t10.

The Delivery Ratio obtained by *IRP* in topics t5 to t7 stabilizes at around 20%, and it is slightly higher than the one obtained by *EP* and *SCORP* from day 6 onwards, while in topics t8 to t10 it stabilizes at around 12%. When compared with the Delivery Ratio obtained by *S&W*, it is much lower. Despite this, *IRP* maintains an increasing trend. This is because *IRP* only delivers messages directly to the interested nodes it encounters, not flooding the network with these messages and saving lots of relays, as it considers them to be uninteresting to the network.

Regarding *S&W*, the Delivery Ratio on the first group of uninteresting topics (t5 to t7) is over 60%, it is higher than the 40% obtained on the second group (t8 to t10). The reason is that at the beginning of the simulation, *S&W* has distributed messages on topics t5 to t10, but during the first eighth days there are no interested nodes in topics t8 to t10, so when nodes start to become interested in t8 to t10, as the buffers have already collapsed, lots of messages have been removed from the network without any opportunity of being delivered. But it has the better results comparing with the other three algorithms.

Analyzing the behavior of *EP* on the first day, its Delivery Ratio in topics t5 to t7 is also high because there are few messages generated and *EP* has not filled its buffers yet, so it has not dropped many messages. Then, during the first eight days, it decreases very quickly from 35% to 12%, because the buffers in *EP* collapsed by forwarding copies of all messages to all nodes and dropping messages that have not yet reached their destination. It should be noted that from day 5 onwards, the result remains stable at 12%. In topics t8 to t10, from day 8 onwards, the Delivery Ratio ranges from 5% to 10%, because when nodes become interested in these topics, their buffers are already collapsed.

The *SCORP*'s Delivery Ratio in the first two days is high in topics t5 to t7, with a maximum of 45% the first day. This is because at that moment there is only interest in those uninteresting topics (and no interest in any other topic) and the few messages generated about them can be relayed and delivered before new nodes interested in other topics appear and fill the buffers.

Then, it decreases to 12%, because, as previously explained, it removes a lot of messages, including the ones on those uninteresting topics, from the buffers of the nodes with a better social metric.

The Delivery Ratio in topics t8 to t10, from day 8 onwards, remains low, ranging from 0% to 5%, because when the interest in those topics starts, nodes have already dropped many messages about them, as there were no interested nodes before day 8, so they were removed from the network even before they were forwarded.

Finally, it can be seen that the Delivery Ratio obtained by *S&W* in uninteresting topics is much higher than the one obtained by the other three algorithms. Anyway, as we said in the previous subsection, the real difference in the total number of messages delivered is very small, as there are few nodes interested in those topics.

7.2.4 Buffer Occupancy

Following, we analyze the Buffer Occupancy, which allows us to compare how efficiently the algorithms manage their buffer.

Figure 7.9 illustrates the daily level of buffer occupancy. It must be noticed that during the first day, *EP* and *S&W* fill their buffers to approximately 100% and 80%, respectively, and they remain full until the end of the simulation. This causes both *EP* and *S&W* start dropping old messages in order to receive the new ones that are generated, and therefore they lose messages that might be of interest later on.

It should be noted that *SCORP* starts to fill its buffer with the messages of the topics that appear and stays filled on average between 75% and 90%. *SCORP* like *EP* and *S&W* does not identify which topics are more interesting than others and therefore fills it with messages from all topics.

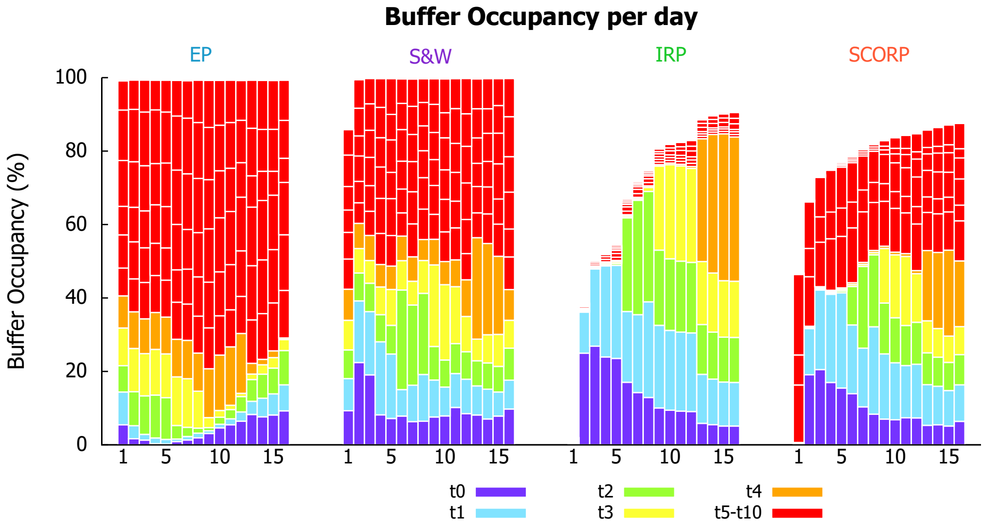


Fig. 7.9: Buffer Occupancy per day of the four analyzed algorithms. Every block represents the percentage of all the nodes' buffers taken up by the messages on every topic. Uninteresting topics (t5 to t10) are represented separately, but all the same color.

IRP's Buffer Occupancy remains below 5% during the first day because there are no interesting topics yet. It does not reach 80% of occupancy until day 10, so it is able to maintain a lot of old messages that could be of interest to someone later.

Every time the interest in a specific topic starts, the amount of buffer devoted by *EP* to the messages in this topic decreases. This happens because the received messages are quickly dropped from the buffer when space for new messages is required and, as we explained in subsection 7.1.4, messages already received by an interested node are not received again, and, therefore, not put in the buffer by the same node anymore.

Concerning *S&W*, the amount of buffer that it devotes to messages on topics that become interesting starts to grow, doubling or tripling the existing number of messages on that topic. This is because messages for which the maximum number of copies (L) had been reached are now delivered by direct transmission to interested nodes, which at the same time place them in their buffers. It can be seen that *S&W* uses approximately the same buffer percentage for each topic with no interest at any moment.

The buffer of *SCORP* is also filled with the messages on interesting and uninteresting topics. While there are no nodes interested in a topic, *SCORP* does not forward messages. Then, when the nodes start to become interest in a topic, the nodes that are socially better connected on every topic fill up their buffers, causing the average buffer occupancy to range from approximately 50% to 90%. All this happens because *SCORP* has no mechanism to detect when to stop forwarding messages that are no longer of interest, so it wastes resources relaying messages on uninteresting topics as well as on interesting ones when there are no more interested nodes.

Regarding *IRP*, it optimizes its buffer usage by learning which topics are of interest and which are not, so, it uses the available resources mainly for interesting topics. For this reason, buffer is mainly not used for a topic while there is no interest in it. Then, when nodes start to become interested in a topic, the Buffer Occupancy in that topic grows, and it only decreases when there is no perception of interest and buffer space is required.

Lastly, it is noteworthy that a significant portion of the *EP*, *S&W*, and *SCORP* buffers are occupied by messages on uninteresting topics, whereas *IRP* effectively manages this resource through interest-based buffer management, which enables it to forward messages only when sufficient interest is detected.

7.2.5 Message Latency

In this subsection, we analyze the Message Latency obtained by the four algorithms in the five interesting topics. We use Equation 4.7, proposed in section 4.2, that only takes into account the time in which the nodes have been interested.

Figure 7.10 shows the Latency that the four analyzed algorithms have obtained in the interesting topics. We can observe that the Latency of *IRP* and *SCORP* increases progressively. At the end of the simulation, *IRP* obtains about 18 hours (10 hours more than *EP* and *S&W* and, 4.4 hours more than *SCORP*). *SCORP* obtains about 14 hours (5.7 hours more than *EP* and *S&W*). The Latency between *EP* and *S&W* is almost the same, it is about 8 hours.

The Latency obtained by *IRP* is the highest. The results are influenced by the mobility model and the time when nodes' interest begins. *IRP* requires nodes to have contact with each other to determine the interest in the network, and these contacts mainly occur during working hours. Therefore, the later in the day the interest in a topic starts (as in the cases of topics t2, t3 and the second period of interest of t1), the less time *IRP* will have to build its perception of interest in the network in that topic, the longer it will take to start delivering messages (because messages are not forwarded until the perception reaches a certain threshold), and the higher the Latency will be.

In the case of *SCORP*, its Latency is slightly lower than that obtained by *IRP*. This is because *SCORP* forwards messages about a topic as soon as any node becomes interested in it, without waiting until the interest reaches a certain threshold to start

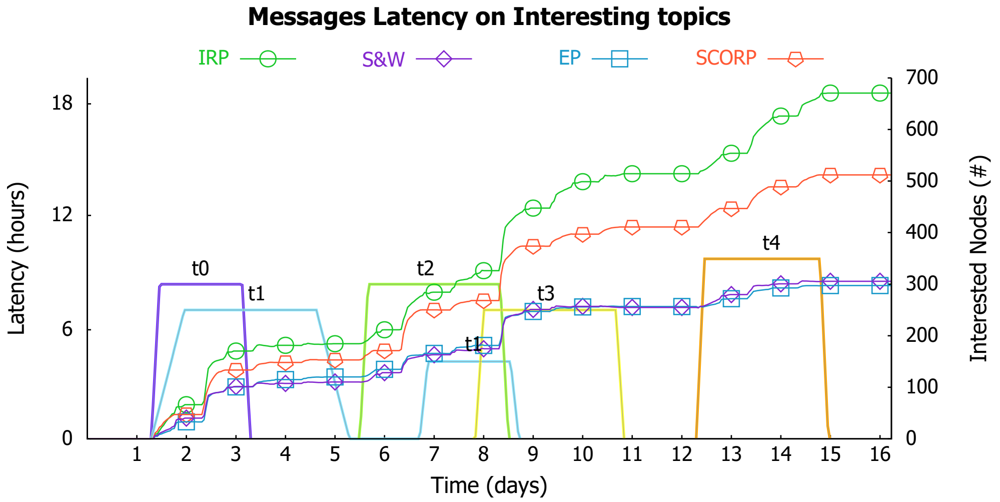


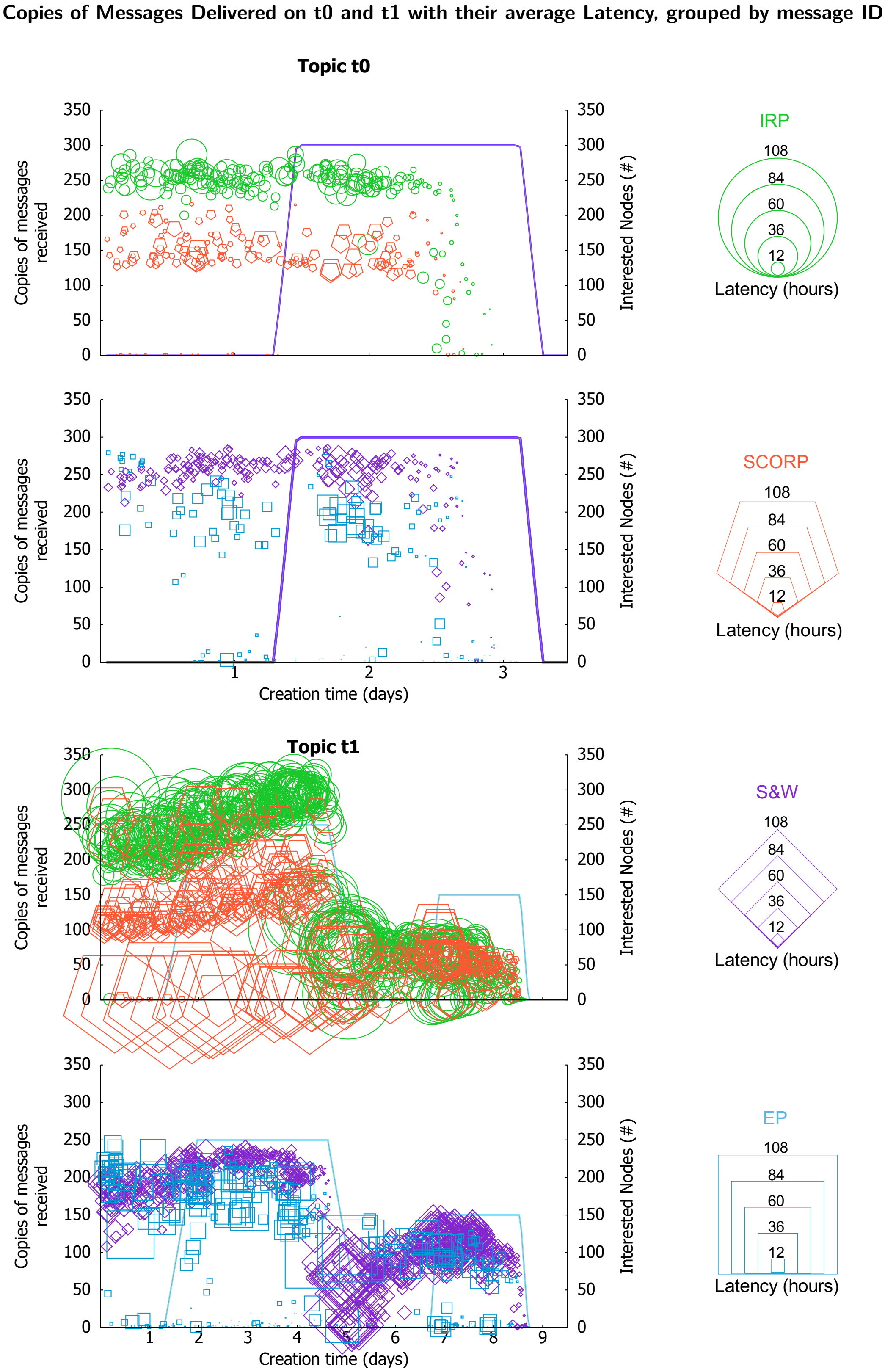
Fig. 7.10: Average Messages Latency on Interesting topics obtained by *IRP*, *S&W*, *EP* and *SCORP* considering the users' interest in the topics.

forwarding (note that this comes at the risk of flooding the network or collapsing the buffers).

The lower Latency of *S&W* and *EP*, with a very little difference between them, is because both have distributed multiple copies of each message before there are interested nodes. This strategy helps them to deliver messages as soon as the nodes become interested. For this reason, both are the least affected by the moments in which the interest in a topic begins. However, by forwarding messages as soon as they are created, even when there are no interested nodes, it causes them to waste resources by flooding the network and collapsing the buffers (Fig. 7.9). In both cases, interesting messages created a while before there are nodes interested in those topics are dropped and, in general, less of them are delivered.

Note that, to the users, the Latency generated by the night hours is not very meaningful, since they usually would not be able to read the messages during that period.

In order to give a more detailed explanation of the results obtained by *IRP*, *S&W*, *EP* and *SCORP*, we show in the bubble charts of Figures 7.11, 7.12 and 7.13 the number of copies of each message that these protocols have delivered (the bubble is located higher or lower) and their average Messages Latency (the bubble is bigger or smaller). Messages are grouped by their message ID, because all copies of the same message have the same message ID, and placed on the timeline (from left to right) according to the original message's creation time. Messages without a single copy delivered are not plotted.



Copies of Messages Delivered on t0 and t1 with their average Latency, grouped by message ID

Topic t0

Topic t1

Fig. 7.11: Number of copies and latency of each delivered messages on topics t0 and t1, grouped by message ID. The number of interested nodes in t0 and t1 is also plotted as a reference in order to help understand the behavior of the different protocols.

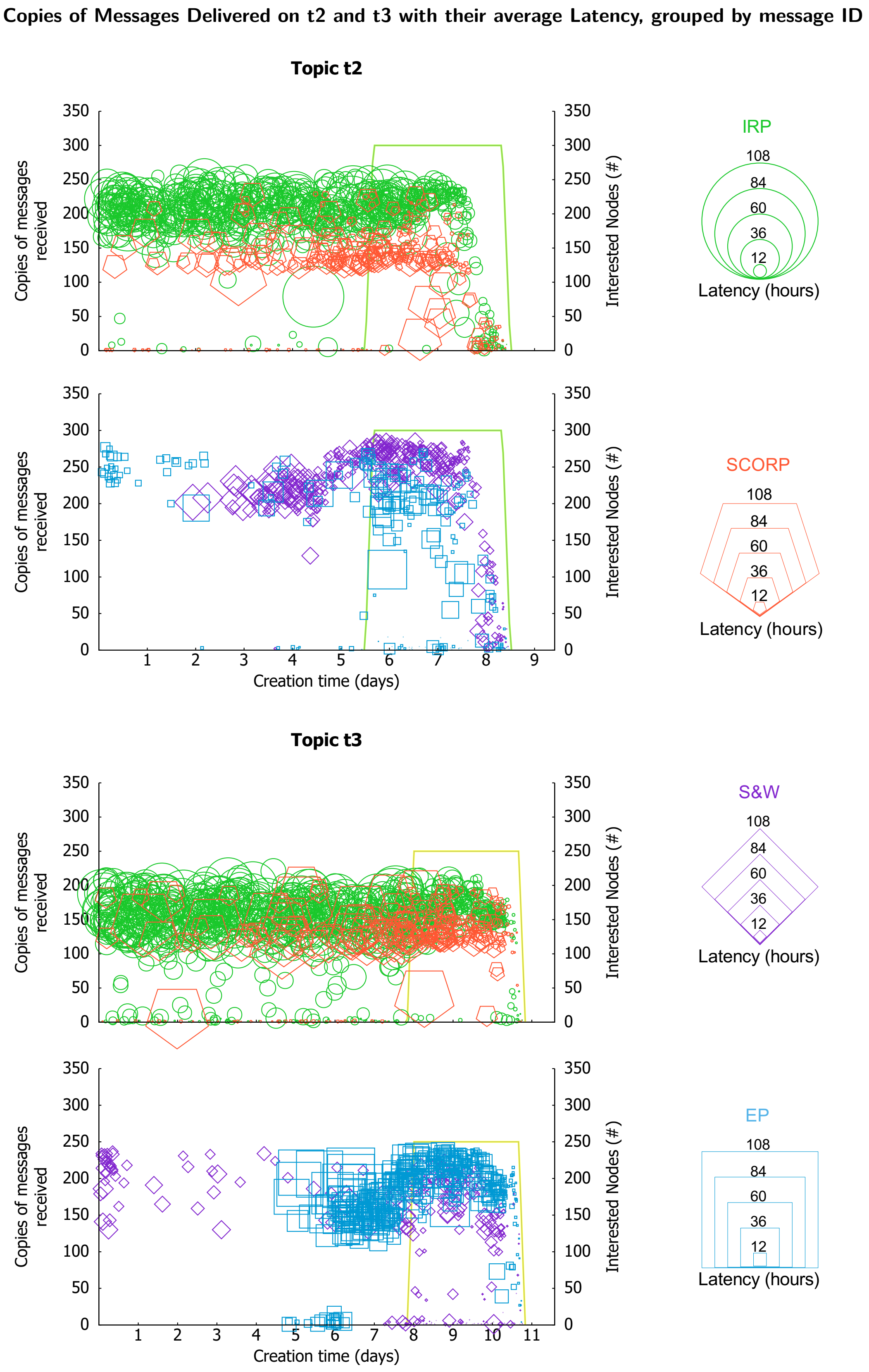


Fig. 7.12: Number of copies and latency of each delivered messages on topics t2 and t3, grouped by message ID. The number of interested nodes in t2 and t3 is also plotted as a reference in order to help understand the behavior of the different protocols.

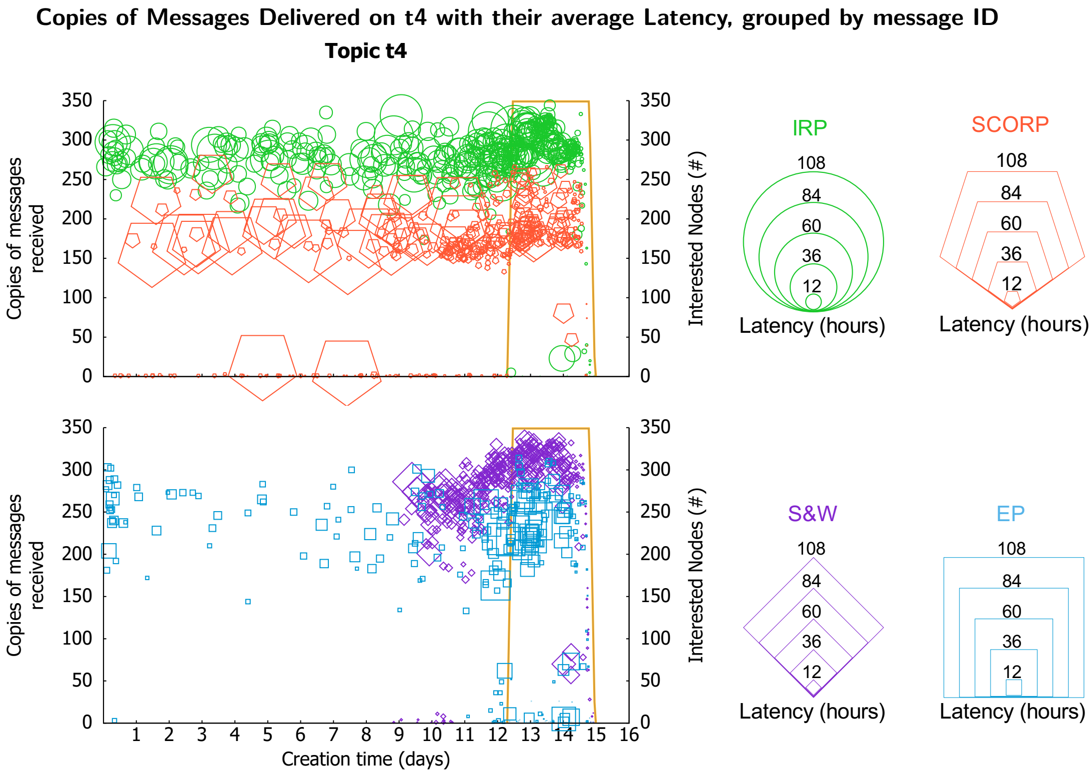


Fig. 7.13: Number of copies and latency of each delivered messages on topic t4, grouped by message ID. The number of interested nodes in t4 is also plotted as a reference in order to help understand the behavior of the different protocols.

We observe in those figures that *IRP* delivers messages generated almost from the beginning of the simulation until almost the end of the interest in each topic (more clearly noticed in topics t3 and t4), which confirms that *IRP* has the ability to keep old messages to deliver them when there is interest in them. However, due to what has been previously explained regarding the topics whose interest starts to build at the evenings or nights, its Latency is the highest.

Although *SCORP* operates with interest like *IRP*, it has no mechanism to stop forwarding messages when there is no longer interest in them, so it eventually continues to forward and occupy its buffers with the messages of the topics in which there has been interest (interesting and uninteresting), and in order to keep them, it removes messages that could be delivered later. We know that *SCORP* has delivered the fewest messages, some of which are messages generated before the interest in the topic begins, many more than *EP*, but many fewer than *IRP* (clearly seen in t3 and t4). So, it also has lower Latency than *IRP*.

It can also be seen that *EP* and *S&W* have the lowest Latencies (smaller bubbles). With these Figures we confirm the fact that they have removed most of the old messages generated at the beginning of the simulation, because they have distributed many messages in the network before there are any interested nodes, collapsing the buffers and dropping them. So, when there is interest in a topic, they deliver the messages they have forwarded over the network and have not discarded yet as soon as interested nodes appear. This can be seen more clearly in topics t3 and t4.

Finally, we analyzed the algorithms' behavior regarding t1, the only topic with two different periods of interest. All algorithms exhibit the highest Latency per message in this topic, as can be seen by comparing the bubble sizes of the t1 bubbles with those of the other topics.

In the case of *IRP* and *SCORP* (mainly in *IRP* and, in a lesser proportion, in *SCORP*), they are able to keep the messages generated before the end of the first period of interest of t1 until the second period of interest and deliver them then. This is signaled by the fact that the total number of delivered copies of the messages generated before the end of the first period of t1 is greater than the number of interested nodes in that period. Obviously, this increases the Messages Latency in this topic. Note that many of the deliveries of these messages could not been done in the first period because some of the nodes interested in the second period were not interested yet, so these messages were impossible to deliver before the second period of interest.

In the case of *S&W* and *EP*, as discussed above, many of the messages generated before and during the first period of interest of t1 are not delivered during the second period of interest of t1 because they have already been removed from their buffers. This is evidenced by the fact that the total number of copies of messages delivered does not exceed the number of interested nodes. And this, together with the small bubbles, indicates that messages generated before the end of the first period of interest of t1 are generally not delivered in the second period of t1.

7.2.6 Efficiency

Finally, we analyze the Efficiency of the four algorithms as the ratio between the Messages Delivered and the Messages Relayed, which allows comparing the resources spent to successfully deliver the messages.

Figure 7.14 shows the Efficiency obtained by the four algorithms on interesting and uninteresting topics.

EP achieves the lowest Efficiency in all topics because it spends too many resources by flooding the network. As we have seen in subsections 7.2.1 and 7.2.2, it forwards millions of messages, but it delivers very few.

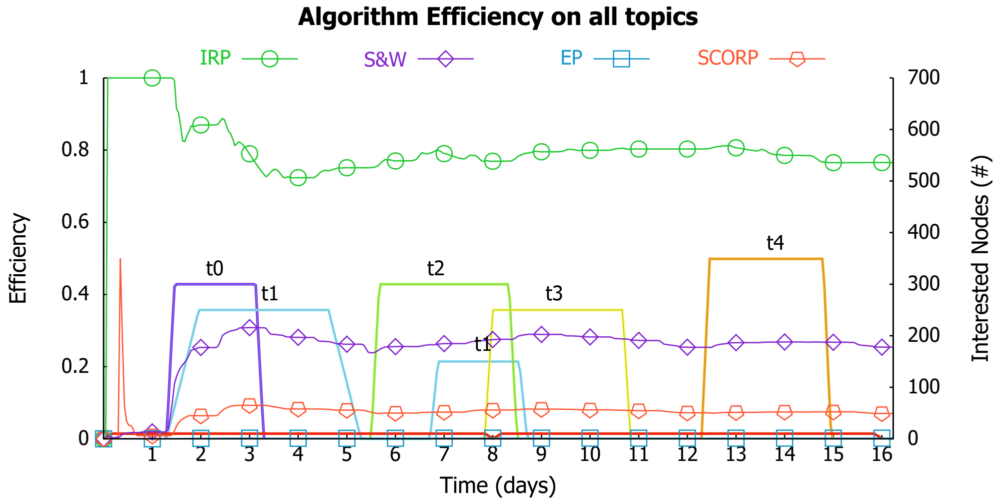


Fig. 7.14: Efficiency of every algorithm as the ratio between Messages Delivered and Messages Relayed in all topics.

The Efficiency obtained by *SCORP* is about one-tenth of that of *IRP* on all topics. This is because it delivers few messages about every topic when there is interest in them, and forwards a lot of messages on these topics. Even when nodes become uninterested in the topics, it continues relaying messages to the better socially connected nodes trying to reach a destination.

The *S&W* Efficiency is one-third of that of *IRP* on all topics. This is because it relays more messages trying to reach the interested nodes, and is only limited by the number of copies it can make (L).

The most efficient algorithm is *IRP* because it limits the number of messages that are forwarded in the network when it considers that there is not enough interest in the area around the nodes. Additionally, this fact does not affect the number of messages delivered, as we have already seen in Subsection 7.2.1 above.

7.3 Summary of the results

In this chapter, we presented a new performance evaluation, and used the redefined metrics presented in Chapter 4 in order to obtain better information by taking into account only the periods in which users are interested.

We performed eleven different topics, and separated them in interesting and uninteresting topics. The interesting were the ones in which there were a high number of interested nodes and the uninteresting, the ones in which there were a low number of interested nodes.

The results shown that *IRP* delivered more messages on interesting topics than the other algorithms, it delivered 7%, 64% and 70% more messages on interesting topics than those delivered by *S&W* and *EP* and *SCORP*, respectively.

IRP reduced relaying of messages on interesting topics by 45.94% compared to *S&W*, 63.03% compared to *SCORP*, and 99.28% compared to *EP*. Similarly, it significantly reduced relays of messages on uninteresting topics by 99.17% compared to *S&W*, 99.50% compared to *SCORP*, and 99.99% compared to *EP*.

In terms of the Delivery Ratio, *IRP* slightly outperformed the Delivery Ratio obtained by *S&W* because it had a delay in starting to forward messages on interesting topics, in contrast to *EP* and *S&W*, which distributed messages immediately after their creation. But, it outperformed the Delivery Ratio obtained by *SCORP* and *EP*. On uninteresting topics, *IRP* performed better than *SCORP* and *EP* but, it had a similar Delivery Ratio compared to *S&W*.

We observed that *IRP* prioritized messages on interesting topics, unlike *EP*, *S&W*, and *SCORP*, which occupied a large amount of buffers with messages on uninteresting topics. So, *IRP* optimized its resources by allocating space primarily to messages on topics of interest, avoiding filling with uninteresting ones.

IRP obtained the highest Latency due to its design, which does not forward messages until there is enough interest in the network. We observed that its Latency was affected by the time at which interest in a topic began, and by the mobility model of the nodes. The later in the day the interest in a topic started (as in the cases of topics t2, t3, and the second period of interest of t1), the less time *IRP* had to build its perception of that topic's interest in the network. As a result, *IRP* started forwarding and delivering messages later than the other algorithms.

The time at which interest in a topic begins, and the mobility model also affected *SCORP*, which like *IRP* must also learn about interest about a topic. However, *EP* and *S&W* were the least influenced by these facts, since, regardless of when the interest in a topic begins, they have already forwarded messages from all topics, and when the interest in one of them begins, they were ready to deliver them.

The most efficient algorithm was *IRP* because it limited the number of messages forwarded in the network when it considered that there was not enough interest in the area around the nodes. Additionally, its Efficiency was not affected by the number of topics that the nodes were simultaneously interested in, nor by the fact that nodes became interested again in a previously interesting topic.

Finally, the overall results obtained by *IRP* in Messages Delivered, Messages Relayed, Buffer Occupancy and Delivery Ratio outperformed the results obtained by *S&W*, *EP* and *SCORP*. *IRP* achieved this performance because it saved resources on uninteresting topics, and then it used them to relay mainly the messages on interesting topics, which were far more important to the users than the uninteresting ones.

Part IV

Conclusions and Future work

Conclusions and Future Lines

8

THE following chapter draws the Conclusions about the design of the Interest-based Routing Protocol (*IRP*), the redefinition of the Delivery Ratio and Latency metrics and the performance evaluation of *IRP*. Finally, we also present some of the Future open lines.

8.1 Conclusions

In this work, we presented two contributions, the fists one was the design and implementation of Interest-based Routing Protocol (*IRP*), a novel routing strategy that use real-time user interest in message topics to intelligently forward messages to anyone interested in a particular topic.

In order to design *IRP*, we introduced two key concepts: *Interest in a topic* and *Perception of Interest in a topic*, that were combined to implement the strategy that allows *IRP* to make forwarding decisions.

The forwarding decision of *IRP* was explained. Nodes first forward the messages on topics of interest to the other node, sorted from highest to lowest perception, and then, messages on topics with high perception, also sorted from highest to lowest perception. In both cases, messages on topics with the same perception are sorted by their age (newer first).

We also described the *IRP* buffer management mechanism, and highlighted that when it needed space, the message dropped was always the oldest message received of the topic (or topics) with the lowest perception. With this design, it was not necessary to set a TTL value in *IRP*, as messages remained in the network as long as there was enough interest in them.

Next, we presented our second contribution, the redefinition of the Delivery Ratio and Latency metrics, to provide a more accurate calculation of them in scenarios with changing user interests. The Delivery Ratio was reformulated by taking into account only the nodes that were interested in receiving a message. In the case of the Latency, it was excluded the time between consecutive periods of interest when the nodes were not interested in receiving messages.

After having presented the contributions of this work. To evaluate the performance of the proposal, we designed several scenarios in which the users' interest changes in real time. Firstly, we used a scenario with a small map of Eixample with random movement, encounters, and interests. In it, we compared the obtained results of *IRP* with two well-known protocols, Spray & Wait (*S&W*) and Epidemic (*EP*).

These results were analyzed and showed the advantages of using *IRP* from the point of view of Messages Delivered and Messages Relayed. It was observed that *IRP* delivered more older messages than the others, so, its Latency was higher compared with them. Additionally, it was possible to see that the Buffer Occupancy changed according to the overall interest in the network, specifically when nodes perceived interest in a particular topic, because the buffer was filled with messages related to that topic.

Then, we included the Social-aware Content-based Opportunistic Routing Protocol (*SCORP*) in a new evaluation by using a scenario with real contact traces to analyze the performance between *IRP*, *EP* and *SCORP* in it. The results demonstrated the effectiveness of *IRP* in delivering messages while minimizing network overhead with multiple peaks of interest. They highlighted the importance of having a mechanism that enabled us to determine when to stop message forwarding due to the loss of interest of the nodes. Unlike *SCORP*, *IRP* detected these changes and leveraged them to forward more interesting messages and thus conserved the energy of the nodes in the network.

Finally, we evaluated the performance of *IRP* by using the redefined Delivery Ratio and Latency metrics in a scenario with the real map of Eixample and its surroundings that simulated a typical working daily routine.

The performance of the four algorithm (*IRP*, *SCORP*, *S&W* and *EP*) were analyzed. The results showed that *IRP*'s Delivery Ratio outperformed the other algorithms studied. The *IRP*'s Latency was slightly higher because before starting to forward and deliver messages, it had to learn about interest, which could delay the start of message forwarding.

The Latency of *IRP* increased due to the time it took for interest to be propagated and reach the network interest threshold. This effect was most noticeable when interest started later in the day. In the case of *SCORP*, once interest in a topic began, it forwarded messages regardless of the network's global interest in that topic, thus delivering messages faster and achieving a slightly lower Latency than *IRP*. *S&W* and *EP* started forwarding messages regardless there is interest or not in the network, flooding it with a lot of messages, allowing them to be delivered as soon as interest in a topic began, resulting in lower Latency compared to *IRP* and *SCORP*.

In summary, in this work, we presented the design of Interest-based Routing Protocol (*IRP*), a new routing strategy that uses users' interest to forward messages to unknown destinations that change in real time according to their own interest. We introduced the concepts of *Interest in a topic* and *Perception of Interest in a topic* as the strategy for forwarding decisions. We explained the *IRP* buffer management and demonstrated its ability to retain messages as long as there is enough interest in them, obviating the need for a TTL field. We redefined the Delivery Ratio and Latency metrics in order to obtain better information about the performance of the algorithms by taking into account the changing interests of users. We designed several experiments, where *IRP* outperformed traditional protocols like *EP* or *S&W* in delivering interesting messages while optimizing network resources and conserving energy. We demonstrated that *IRP*, allowed the real-time changes in users' interests on topics in order to deliver as many interesting messages as possible. And, that it improved the use of network resources, relaying only when and where there were interest.

8.2 Future Lines

Once the objectives of this work have been accomplished, we propose the following future lines of action.

IRP has been designed to operate well when there are many nodes interested in a topic. This fact makes the performance on uninteresting topics (when there are few interested nodes) low. In the future, it would be interesting to allow *IRP* to automatically adjust the perception thresholds on these topics according to the environment that each node perceives.

In this work, a normal and trapezoidal distribution to model the interest of the nodes to evaluate the performance of the algorithms were used. It would be interesting to evaluate the performance of *IRP* with a new model of changing interests, obtaining this information directly from real applications.

The experiments used messages in which only one topic was associated per message. Therefore, allowing messages to be associated with several topics and received by nodes interested in any of them is a potential opportunity to design a new protocol, or adapt existing ones, to support this functionality.

Finally, developing a user application that has an impact like the current applications on the Internet for motivating users to participate in these networks in a secure environment to them, is still a pending challenge.

Part V

Bibliography

Bibliography

[1] Majeed Alajeely, Robin Doss, and Asma'a Ahmad. "Routing Protocols in Opportunistic Networks – A Survey". In: *IETE Technical Review* 35.4 (2018), pp. 369–387 (cit. on pp. 3, 9).

[2] Aref Hassan Kurd Ali, Halikul Lenando, Slim Chaoui, Mohamad Alrfaay, and Medhat A. Tawfeek. "A Dynamic Resource-Aware Routing Protocol in Resource-Constrained Opportunistic Networks". In: *Computers, Materials & Continua* 70 (2 2022), pp. 4147–4167 (cit. on p. 11).

[3] Vittalis Ayu, Bambang Soelistijanto, and Junandus Sijabat. "Interest-based Epidemic Routing in Opportunistic Mobile Networks". In: *2019 7th International Conference on Information and Communication Technology (ICoICT)*. 2019, pp. 1–6 (cit. on p. 11).

[4] Raphael Bialon and Kalman Graffi. "Congestion Control for Epidemic Routing in Opportunistic Networks". In: *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2019, pp. 102–109 (cit. on p. 10).

[5] Ying Cai, Haochen Zhang, Yanfang Fan, and Hongke Xia. "A survey on routing algorithms for opportunistic mobile social networks". In: *China Communications* 18 (2 Feb. 2021), pp. 86–109 (cit. on p. 12).

[6] Radu-Ioan Ciobanu, Radu-Corneliu Marin, Ciprian Dobre, Valentin Cristea, and Constandinos X. Mavromoustakis. "ONSIDE: Socially-aware and Interest-based dissemination in opportunistic networks". In: *2014 IEEE Network Operations and Management Symposium (NOMS)*. 2014, pp. 1–6 (cit. on pp. 13, 15).

[7] Marco Conti, Chiara Boldrini, Salil S Kanhere, et al. "From MANET to people-centric networking: Milestones and open research challenges". In: *Computer Communications* 71 (2015), pp. 1–21 (cit. on p. 9).

[8] Daniela Cordova-Pintado, Adrian Sanchez-Carmona, and Ramon Marti. "Paving the way towards delivering messages based on user interests in OppNets". English. In: *2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. Oct. 2022, pp. 190–197 (cit. on p. 39).

[9] Daniela Córdova-Pintado, Adrián Sánchez-Carmona, and Ramon Martí. "Interest-based Routing in Opportunistic Networks: Evaluating IRP against SCORP". In: *2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2023, pp. 188–193 (cit. on p. 51).

[10] Xia Deng, Le Chang, Jun Tao, and Jianping Pan. “Reducing the Overhead of Multicast Using Social Features in Mobile Opportunistic Networks”. In: *IEEE Access* 7 (2019), pp. 50095–50108 (cit. on p. 12).

[11] J. Dorp and Samuel Kotz. “Generalized trapezoidal distributions”. In: *Metrika* 58 (Aug. 2003), pp. 85–97 (cit. on p. 52).

[12] Nathan Eagle and Alex (Sandy) Pentland. *CRAWDAD mit/reality*. 2022 (cit. on p. 51).

[13] Frans Ekman, Ari Keränen, Jouni Karvo, and Jörg Ott. “Working day movement model”. In: ACM Press, 2008, p. 33 (cit. on p. 64).

[14] Vijay Erramilli, Mark Crovella, Augustin Chaintreau, and Christophe Diot. “Delegation Forwarding”. In: May 2008, pp. 251–260 (cit. on p. 11).

[15] Diego Freire, Sergi Robles, and Carlos Borrego. “Towards a Methodology for the Development of Routing Algorithms in Opportunistic Networks”. In: *arXiv preprint arXiv:2009.01532* (2020) (cit. on p. 65).

[16] W. Gao, Q. Li, B. Zhao, and G. Cao. “Social-Aware Multicast in Disruption-Tolerant Networks”. In: *IEEE/ACM Transactions on Networking* 20.5 (2012), pp. 1553–1566 (cit. on p. 12).

[17] Wei Gao, Wenjie Hu, and Guohong Cao. “Interest-Based Data Dissemination in Opportunistic Mobile Networks: Design, Implementation, and Evaluation”. In: Aug. 2014, pp. 215–237 (cit. on p. 13).

[18] Pan Hui, Augustin Chaintreau, James Scott, et al. “Pocket switched networks and human mobility in conference environments”. In: *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. 2005, pp. 244–251 (cit. on p. 9).

[19] Pan Hui, Jon Crowcroft, and Eiko Yoneki. “BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks”. In: *IEEE Transactions on Mobile Computing* 10.11 (2011), pp. 1576–1589 (cit. on p. 13).

[20] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. *The ONE Simulator for DTN Protocol Evaluation*. Rome, Italy, 2009 (cit. on pp. 4, 39, 51, 64).

[21] Khuram Khalid, Isaac Woungang, Sanjay K. Dhurandher, Jagdeep Singh, and Leonard Barolli. “A fuzzy-based check-and-spray geocast routing protocol for opportunistic networks”. In: *Journal of High Speed Networks* 27 (1 Mar. 2021), pp. 1–12 (cit. on p. 10).

[22] Khuram Khalid, Isaac Woungang, Sanjay Kumar Dhurandher, Jagdeep Singh, and Joel J. P. C. Rodrigues. “Energy-Efficient Check-and-Spray Geocast Routing Protocol for Opportunistic Networks”. In: *Information* 11.11 (2020) (cit. on p. 10).

[23] Prashant Kumar, Naveen Chauhan, and Narottam Chand. “Security Framework for Opportunistic Networks”. In: 2018, pp. 465–471 (cit. on p. 29).

[24] Halikul Lenando and Mohamad Alrfaay. “EpSoc: social-based epidemic-based routing protocol in opportunistic mobile social network”. In: *Mobile Information Systems* 2018 (2018) (cit. on p. 10).

[25] Na Li and Sajal K. Das. “A trust-based framework for data forwarding in opportunistic networks”. In: *Ad Hoc Networks* 11 (4 June 2013), pp. 1497–1509 (cit. on p. 29).

[26] Anders Lindgren, Avri Doria, and Olov Schelén. “Probabilistic Routing in Intermittently Connected Networks”. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 7.3 (July 2003), pp. 19–20 (cit. on p. 10).

[27] Abraham Martín-Campillo, Jon Crowcroft, Eiko Yoneki, and Ramon Martí. “Evaluating opportunistic networks in disaster scenarios”. In: *Journal of Network and Computer Applications* 36.2 (2013), pp. 870–880 (cit. on p. 3).

[28] Alessandro Mei, Giacomo Morabito, Paolo Santi, and Julinda Stefa. “Social-aware stateless forwarding in pocket switched networks”. In: IEEE, Apr. 2011, pp. 251–255 (cit. on p. 13).

[29] Waldir Moreira, Paulo Mendes, and Susana Sargento. “Social-Aware Opportunistic Routing Protocol Based on User’s Interactions and Interests”. In: 2014, pp. 100–115 (cit. on pp. 4, 13, 15).

[30] Vinícius F.S. Mota, Felipe D. Cunha, Daniel F. Macedo, José M.S. Nogueira, and Antonio A.F. Loureiro. “Protocols, mobility models and tools in opportunistic networks: A survey”. In: *Computer Communications* 48 (July 2014), pp. 5–19 (cit. on p. 9).

[31] Elena Pagani and Gian Rossi. “Interest-driven forwarding for delay-tolerant mobile ad hoc networks”. In: July 2013, pp. 718–723 (cit. on pp. 13, 15).

[32] Noelia Pérez Palma, Falko Dressler, and Vincenzo Mancuso. “Precise: Predictive Content Dissemination Scheme exploiting realistic mobility patterns”. In: *Computer Networks* 201 (2021), p. 108556 (cit. on p. 11).

[33] Aydin Rajaei, Dan Chalmers, Ian Wakeman, and George Parisis. “Efficient Geocasting in Opportunistic Networks”. In: *Computer Communications* 127 (2018), pp. 105–121 (cit. on p. 16).

[34] Nikodin Ristanovic, George Theodorakopoulos, and Jean-Yves Le Boudec. “Traps and pitfalls of using contact traces in performance studies of opportunistic networks”. In: *2012 Proceedings IEEE INFOCOM*. 2012, pp. 1377–1385 (cit. on p. 3).

[35] Adrián Sánchez-Carmona, Frédéric Guidec, Pascale Launay, Yves Mahéo, and Sergi Robles. “Filling in the missing link between simulation and application in opportunistic networking”. In: *Journal of Systems and Software* 142 (2018), pp. 57–72 (cit. on p. 16).

[36] Adrián Sánchez-Carmona, Sergi Robles, and Carlos Borrego. “Endeavouring to be in the good books. Awarding DTN network use for acknowledging the reception of bundles”. In: *Computer Networks* 83 (June 2015), pp. 149–166 (cit. on p. 3).

[37] Jagdeep Singh, Sanjay Kumar Dhurandher, and Vinesh Kumar. “Mobility Models in Opportunistic Networks”. In: *Opportunistic Networks*. CRC Press, 2021, pp. 225–242 (cit. on p. 40).

[38] Libo Song and David F. Kotz. “Evaluating Opportunistic Routing Protocols with Large Realistic Contact Traces”. In: *Proceedings of the Second ACM Workshop on Challenged Networks*. CHANTS ’07. Montreal, Quebec, Canada: Association for Computing Machinery, 2007, pp. 35–42 (cit. on p. 16).

[39] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. “Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks”. In: WDTN ’05. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 2005, pp. 252–259 (cit. on pp. 4, 10, 42, 66).

[40] NN Srinidhi, CS Sagar, Chethan S Deepak, J Shreyas, and Kumar SM Dilip. “An improved PRoPHET - Random forest based optimized multi-copy routing for opportunistic IoT networks”. In: *Internet of Things* 11 (2020), p. 100203 (cit. on p. 10).

[41] Asanga Udugama, Jens Dede, Anna Förster, et al. “My Smartphone tattles: Considering Popularity of Messages in Opportunistic Data Dissemination”. In: *Future Internet* 11.2 (2019) (cit. on pp. 13, 15).

[42] Amin Vahdat, David Becker, et al. “Epidemic routing for partially connected ad hoc networks”. In: *Technical Report CS-200006, Duke University* (2000) (cit. on pp. 4, 10).

[43] Hanno Wirtz, Jan Rüth, Torsten Zimmermann, and Klaus Wehrle. “Interest-Based Cloud-Facilitated Opportunistic Networking”. In: *Proceedings of the 8th ACM MobiCom Workshop on Challenged Networks*. CHANTS ’13. Miami, Florida, USA: Association for Computing Machinery, 2013, pp. 63–68 (cit. on pp. 13, 15).

[44] Yong Xi and Mooi Choo Chuah. “An encounter-based multicast scheme for disruption tolerant networks”. In: *Computer Communications* 32 (16 Oct. 2009), pp. 1742–1756 (cit. on p. 12).

[45] Sui Yu, Lichen Zhang, Lixia Li, et al. “An Efficient Interest-aware Data Dissemination Approach in Opportunistic Networks”. In: *Procedia Computer Science* 147 (2019). 2018 International Conference on Identification, Information and Knowledge in the Internet of Things, pp. 394–399 (cit. on p. 13).

[46] D. Zhang, H. Ma, and D. Zhao. “Social-Aware Backbone-Based Multicast Routing in Mobile Opportunistic Networks”. In: *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*. 2017, pp. 31–38 (cit. on p. 12).

[47] Junbao Zhang, Haojun Huang, Geyong Min, Wang Miao, and Dapeng Wu. “Social-Aware Routing in Mobile Opportunistic Networks”. In: *IEEE Wireless Communications* 28.2 (2021), pp. 152–158 (cit. on p. 12).

María Daniela Córdova
Pintado
Bellaterra, February 2024