

Information-Centric Networking in Next-generation Communications Scenarios

Alessandro Morelli¹, Mauro Tortonesi^{1,*}, Cesare Stefanelli¹, Niranjan Suri^{2,3}

- 1) Department of Engineering
University of Ferrara
Ferrara, Italy
{alessandro.morelli,mauro.tortonesi,cesare.stefanelli}@unife.it
- 2) US Army Research Laboratory
Adelphi, MD, USA
niranjan.suri.civ@mail.mil
- 3) Florida Institute for Human & Machine Cognition
Pensacola, FL, USA
nsuri@ihmc.us

*) Corresponding author:

Name: Mauro Tortonesi
Address: Department of Engineering
University of Ferrara
Via Saragat, 1
44122 Ferrara (FE)
Italy
Phone: +39 0532 974888
Mobile: +39 328 9895530
Fax: +39 0532 974888
E-Mail: mauro.tortonesi@unife.it

Abstract

Next-generation networking environments, characterized by the overlapping of wireless networks of different types, are emerging as a new and extremely interesting scenario. Their high dynamicity and heterogeneity present significant challenges from the communications perspective, which call for the adoption of new paradigms based on opportunistic and Information-Centric Networking (ICN) approaches. Applications operating in next-generation environments have peculiar characteristics that could benefit from ICN-based middleware solutions. This paper presents ICeDiM, a middleware we designed for ICN communications in next-generation scenarios, which builds on top of the innovative concept of *Application-level Dissemination Channels (ADCs)* with tunable permeability levels. A thorough and in-depth experimental evaluation of ICeDiM in a next-generation environment realistically simulated using ICeONE, a modified version of the well-known ONE simulator, demonstrates that our approach can achieve very good performance levels in terms of delivery ratio and network resource consumption.

Keywords: Information-Centric Networking; Next-generation Communications Scenarios; Heterogeneous Networks; Information Dissemination Channels; Network Simulation

1. Introduction

Next-generation networking environments are emerging as a new and extremely interesting communications scenario. Characterized by the overlapping of wireless networks of different types (such as heterogeneous cellular networks with macro, micro, and femto cells, Wi-Fi networks, wireless sensor and actuator networks, etc.), they present very interesting opportunities for the optimization of communication resource usage. Examples include the “offloading” of heavy traffic to the most convenient communication links and the development of location-, social-, and context-aware applications and services [1].

However, next-generation networks present significant challenges from the communications perspective. Their high dynamicity and heterogeneity, with frequent and severe fluctuations in channel conditions, node mobility, disconnected operations, and network partitioning not being the exception but the rule, make the adoption of end-to-end communications particularly challenging [2]. The need to support the resource sharing requirements of location-, social-, and context-aware applications suggests the rethinking of the programming paradigms for communications, promoting the adoption of novel solutions, such as the opportunistic paradigm.

More specifically, next-generation communications scenarios seem a particularly good fit for the design principles of *Information Centric Networking (ICN)*. In ICN, communications do not follow the direct end-to-end paradigm typically used by today’s Internet applications, but depend instead on the exchange of *Information Objects (IO)* through indirect store-and-forward mechanisms. IOs are univocally and globally addressable through their unique name (or identifier) and can be retrieved by means of a request/reply communication model. Applications can request a specific IO or state their interest towards all IOs of a given type.

In next-generation environments, ICN can naturally take advantage of multiple communication links on top of heterogeneous and overlapping networks, support multi-party communications, and enable efficient resource utilization through IO caching. In addition, ICN allows low-overhead local communications, which are essential for implementing social-, location-, and context-aware applications. This makes ICN a very promising paradigm to tackle the main problems arising in next-generation communications scenarios [5].

A very active and growing research community is still working on the major challenges faced by ICN, namely routing, naming, efficient caching, control overhead minimization, etc. Most of those studies have focused so far on the dissemination of long-lived content in static network topologies [3] [6] [7] [8], with implementations like DONA [9], CCN (<http://blogs.parc.com/ccnx/>) [10], NDN (<http://named-data.net/>) [11], 4WARD (<http://www.4ward-project.eu/>) [12], SAIL (www.sail-project.eu), PSIRP (<http://www.psirp.org/>), and its continuation PURSUIT (www.fp7-pursuit.eu) [13]. However, the smart exploitation of the usually scarce and heterogeneous communication resources, typical of next-generation scenarios, has not yet been explored extensively by the research literature on ICN.

This paper aims to reduce that gap: it investigates the feasibility of the ICN paradigm in next-generation communications scenarios by analyzing advantages and difficulties that arise when developing an ICN-based information dissemination middleware. Additionally, our work proposes new concepts and tools specifically targeted to support information dissemination in highly dynamic, resource challenged, and heterogeneous communication environments.

More specifically, this manuscript presents 4 main contributions. First, we analyze the nature of information exchanged in next-generation networks and propose an information and a communication model that represent a guideline for the development of ICN-based solutions in this environment from the exploitation of broadcast communications, information prioritization, and cache management perspectives. Second, we introduce *Application-level Dissemination Channels (ADCs)* with tunable permeability levels, a novel fundamental concept that allows exchange of IOs within application-level thematic channels while, at the same time, enabling control over the node-level commitment to support communications within each channel. By regulating resource consumption and the nodes' collaboration in the effort of exchanging information, ADCs represent a particularly effective building block for ICN-based communications in next-generation environments. Third, we introduce ICeONE, a modified version of the well-known ONE simulator that represents the state of the art for opportunistic networking research. We specifically designed and implemented ICeONE to enable the simulation of ICN-based communication solutions in next-generation environments. Finally, we present ICeDiM, a middleware that we designed and developed building on top of the models and concepts discussed above. We conducted a thorough and in-depth experimental evaluation of ICeDiM in a next-generation environment realistically simulated using ICeONE. The results we obtained show how the models and concepts we devised can keep resource consumption on the network nodes under control while also increasing the delivery ratio of IOs.

2. The Next-generation Communications Scenario and ICN

Next-generation communications scenarios, such as the one depicted in Fig. 1, are characterized by multiple overlapping wireless networks with significantly different administrative and/or channel characteristics. In these scenarios, users (and/or system components) can dynamically switch between multiple network access points, assign different traffic to different networks, or even use them in parallel. This trend is already clearly visible in the emergence of *heterogeneous networks* in LTE/4G infrastructures [14] [15] and in the consequent interest towards device-to-device (D2D) communications [16] [17].

Next-generation scenarios are also becoming more and more common in many modern environments such as smart cities. In fact, while smart cities will have a connected core where the collected data is stored and processed through Cloud-based analytics [18], a large part of the value of their infrastructures will reside at the edge of the network, where a plethora of smart objects, sensors, vehicles, and people occasionally connect to the network through different communication technologies, generating raw data and accessing (often proximity-based) services [19]. The growing interest towards this environment is also demonstrated by the recently emerged “fog computing” research, which investigates methodologies and tools for moving computational resources, storage, and entities from the core of the network to its edge [20] [21].

Building smart applications and services for next-generation networks is a very challenging task [1]. In fact, a multitude of different applications with very different requirements and priorities will run in this environment, and they will have to compete for the sometimes-scarce bandwidth resources and cope with the challenging problems that arise from node mobility, intermittent links, and network partitioning. In addition, applications will often exhibit new and radically different communication patterns, which makes their development extraordinarily arduous when using traditional end-to-end paradigms as the basic building blocks for communications.

The strong correlation between spatial and temporal proximity of users and the information they would be interested in accessing suggests the opportunity to optimize the local consumption of information. In fact, a significant portion of the data exchanged in this scenario will be processed and consumed locally, due to the mostly proximity-based nature of next-generation applications and services, such as social- and location-aware multiplayer games, local- and context-aware interactions with people, information, and services such as transportation. This is especially true considering the increasing availability and pervasiveness of portable devices among the population and the exponential growth of traffic produced by mobile users [22]. Instead of conveying all data using often overburdened communication infrastructures, e.g., the cellular network, applications and services should favor constrained information dissemination and direct communications between nodes, while also leveraging mobile offloading and relaying techniques [17]. Moreover, studies suggest that mobile offloading might end up being significantly more convenient, from the economic point of view, for both end users and service

providers [23].

In the last decade, researchers have developed sophisticated middleware solutions capable of maintaining reliable end-to-end communications in wireless environments in spite of node mobility and temporary link disruption [24] [25] and to make them work seamlessly with existing applications [26] [27]. However, in the presence of high churn and multiple link types, middleware solutions based on the indirect communications paradigm, such as store-and-forward, have been shown to achieve better performance with lower computational overhead [2] [4].

Therefore, next-generation application and service developers would be very well served by communication solutions based on *Information Centric Networking (ICN)* - a recently emerged communication paradigm based on the publish-subscribe model that aims to shift the focus of the Internet from hosts and servers to the content that users want to access [6]. In ICN, Information Objects (IOs) represent the basic unit of information that nodes exchange during communication. A node that wants to retrieve some specific content sends an interest message that is distributed across the network and processed by all nodes it traverses. Any node that has the requested content, e.g., in its cache, can forward the IO to the requestor, effectively serving as a rendezvous point between information producer(s) and consumer(s)¹. By decoupling the communication along both the dimensions of time and space, ICN allows applications to publish and notify their interest in IOs at any moment in time. In this way, the communication is asynchronous and there is no need to establish and maintain an end-to-end connection between content requestor and provider.

ICN-based applications interested in retrieving specific content can do that through the name of the corresponding IO, and do not require any knowledge about the location of the information. A location-independent object identifier (ID) univocally identifies each IO [6]. Research often proposes the use of human-readable hierarchical object IDs, to facilitate the generation of IO requests and increase names' readability and understandability [3] [5].

Name-based IO retrieval also can, and typically does, take advantage of IO replicas to reduce the latency in information access. As any node that has the desired IO can serve as the requestor, ICN-based applications can seamlessly reap the opportunities provided by information locality to optimize communications.

In addition, the in-network caching functionality of ICN, which automatically stores IO replicas in router nodes along the communication path, enables disconnected/delay-tolerant communications and significantly reduces the impact of network partitioning. By caching the "trendiest" IOs in nodes that are located closer to subscribers, or at nodes that have high centrality within the network, ICN increases the number of neighboring nodes that are able to serve requests for the most common or in-demand content [32] [33]. So, even with network partitioning, where no routes from a subscriber to one or more publishers are available, it might

¹ In ICN terminology, nodes that can store content and serve the requests carried within an interest message are called router nodes.

still be possible to retrieve the requested content from other nodes within the reachable portion of the network. Similarly, in-network caching can also provide a higher resilience to link intermittency, as it naturally enables subscribers to request content over multiple links at the same time [5].

Finally, the content-centric perspective of ICN naturally enables the concurrent transfer of IOs from multiple sources and/or to multiple destinations as well as the retrieval of different IOs (or even different portions of the same IO) from multiple nodes at the same time [32]. The result from the users' point of view is an increase in performance, while the network can benefit from the distribution of traffic across multiple sources and paths and, consequently, from a fairer usage of the available bandwidth.

The characteristics of the ICN paradigm make it particularly well suited to support communications middleware that can take advantage of the wide range of network interfaces integrated into modern mobile devices, which varies from short-range, low-power communications, e.g., IEEE 802.15.4 or Bluetooth LE, to medium range communications, e.g., Wi-Fi and WiMax, to expensive, long-range communications, e.g., LTE/4G/5G and satellite [5] [28] [29]. Next-generation applications will thus be able to keep operating seamlessly on mobile devices as the underlying ICN-based middleware exploits the best connection or even uses multiple links at the same time, exploiting features of the ICN stack that allow the middleware to naturally switch connectivity between multiple links to retrieve the requested IOs. Conversely, their counterparts based on direct communications would struggle to keep connections to specific destinations open in presence of node mobility and frequent transmission errors that hinder the preservation of end-to-end connections, and would thus need to invest significant resources to implement the continuous monitoring and probing of all active connections [30] [31].

IOs can carry diverse content: videos, images, songs, e-mails, street directions, scheduled appointments, eHealth data, police reports, etc.; depending on the type of content IOs carry, their size can vary significantly. In next-generation networking scenarios such as Smart Cities, a significant part of the traffic produced will be generated by the Internet of Things (IoT), which provides tools to retrieve data from sensor nodes and control actuators to enable interaction with the environment. IOs could carry sensor data, for instance in the form of XML or JSON structures, and support communications between smart objects and applications that want to access those data or interact with the environment. The presence of metadata fields in IOs [6], such as timestamps, a set of GPS coordinates, tagging, and security-related information, e.g., a signature for accountability purposes, could further ease the retrieval of information based on its age, location, and trust.

In addition, an increasing number of applications depend on the production, dissemination, and consumption of multimedia content, such as social media applications that enable people to broadcast live videos within a circle of people or browse and watch videos from a database with billions of entries. Let us consider, for instance, a person (Alice) sitting in a coffee shop during her lunch break and watching videos on her smart device. The night before, her favorite team

won the final of the soccer World Championship and so Alice wants to re-experience the most exciting moments of the match by watching the highlights. At this point, her ICN-based smartphone notifies the nodes in its proximity of its interest in the IOs carrying last night's game highlights, for example using WiFi. Since the coffee shop is very crowded during lunchtime and due to the match's popularity, two other people in the shop are watching/have just watched the same video on their ICN-enabled tablets. Thanks to in-network caching and the ability to retrieve IOs from multiple sources, those devices can directly respond to the interest message sent by Alice's smartphone and become providers for the requested IO. Once Alice's device has received the IO, it can decide whether to store the IO in its cache and become a potential future provider for the same content.

Unfortunately, while ICN seems to fit very well into the operating context of next-generation communications scenarios, it was essentially devised for the wired Internet. As a result, the methodologies and tools adopted by the most common ICN implementations are not very well suited for wireless communication environments and require substantial modifications to perform effectively in next-generation environments. More specifically, there is a need for new concepts and tools specifically designed for next-generation scenarios that build on top of well-suited information and communication models.

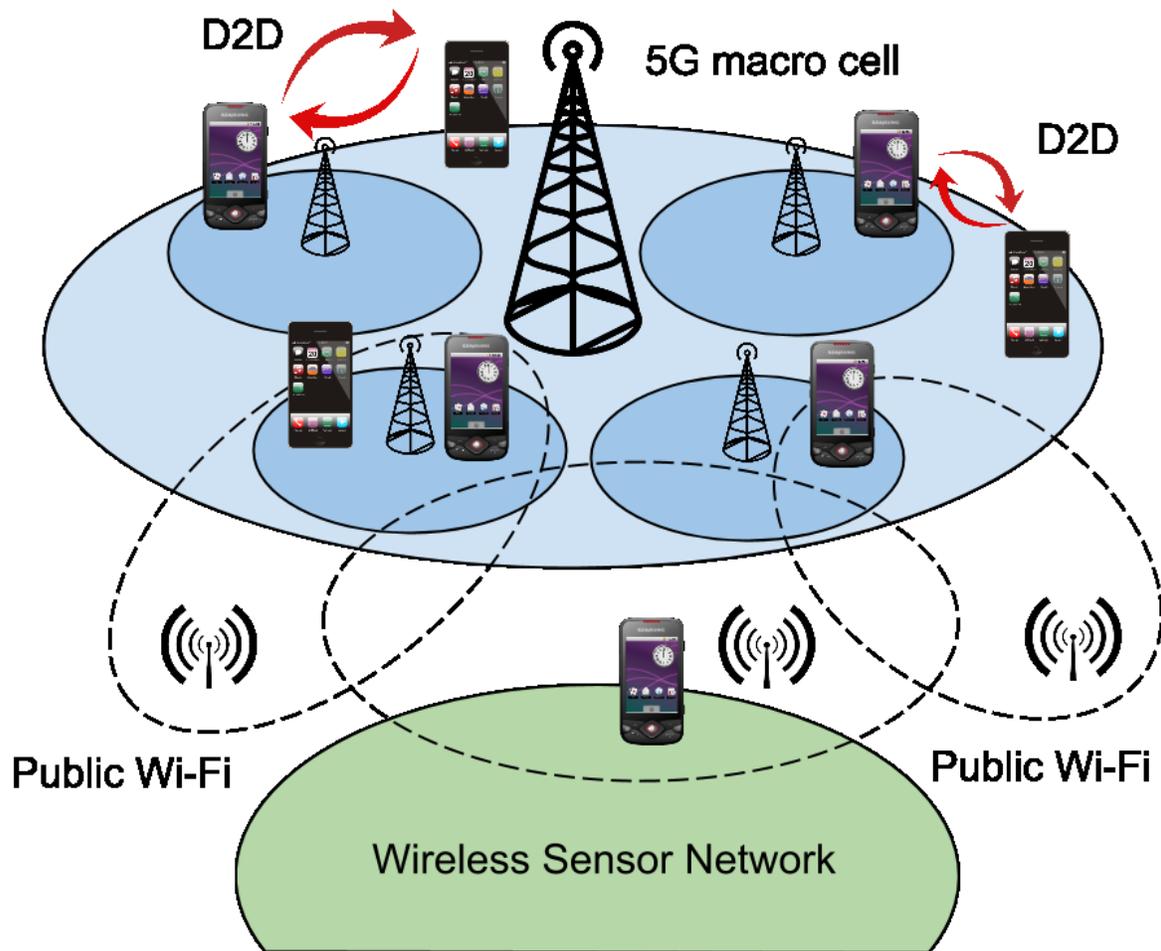


Fig. 1. The next-generation communications scenario.

3. Related Work

Researchers have dedicated significant attention to ICN recently. Several works, including those focused on naming and security, produced results that are in large part applicable to next-generation scenarios, and thus we do not consider them here [6] [34] [35] [36]. Instead, we review the literature on topics that are of great interest for the issues arising in next-generation communications environments. We broadly classify the proposals in three categories: *routing and forwarding*, *caching*, and *complete frameworks*.

3.1. Routing and Forwarding

One of the main concerns encompassing all ICN implementations is routing, a broad topic which can be further decomposed in two subcategories: routing of interest messages and forwarding of data back to the subscriber. While researchers have devised several routing algorithms for ICN, those solutions highly depend on the naming system used and they assume a backbone of router nodes equipped with large, fast storage modules and dedicated processing power to

support caching and scalability [7] [8] [37]. However, such assumptions cannot hold in next-generation communication environments, where network heterogeneity and the major presence of portable devices and sensors cannot provide the necessary memory and computational resources.

From this point of view, research studies on routing algorithms for Delay-Tolerant Networks (DTNs) and on Opportunistic Networking have yielded particularly interesting results. DTNs share many characteristics with the next-generation communications scenario: routing solutions designed for DTNs have to deal with network partitioning, link intermittency, node mobility, limited resources at their disposal [38], and with a highly heterogeneous population of devices [39]. Opportunistic Networking further extends the spontaneous networking concepts developed in the mobile ad hoc and in the DTN research areas by analyzing and exploiting social interaction aspects in human communications [40], human mobility patterns [41], contextual information [42], and even information content [43] in the exchanged messages. Due to its potential, several research efforts proposed Opportunistic Networking as the communication paradigm for many challenging communication environments, from tactical edge networks [2], to disaster recovery scenarios [44], to smart cities, and to deep-space communications, where network infrastructure is not (always) available [45] [46].

The literature on routing algorithms for Opportunistic Networking usually classifies them in two main classes: single-copy and multiple-copy [47]. In the former case, there is always a single carrier for each message; in the multiple-copy case, instead, messages can be duplicated in order to increase their availability throughout the network and exploit multiple paths to reach destination(s) more efficiently. Single-copy routing algorithms can still use broadcast transmission to deliver some content to the subscriber so that neighbors can increase their knowledge of the state of the network by processing broadcasted packets. However, the only way to take advantage of content dissemination-based caching strategy is by using multiple-copy routing algorithms.

Routing algorithms for Opportunistic Networking can be further classified depending on the type of knowledge on which they base the routing decision-making process. In accordance with [48], we distinguish context-oblivious, mobility-based, and social context-based algorithms. Algorithms like Epidemic Routing (ER) [49] and Spray and Wait (SnW) [50] belong to the first category, P_{Ro}PHET [51], MaxProp [52], and Spray and Focus (SnF) [53] are examples of mobility-based algorithms, and finally HiBOp [54], SpatioTempo [55], ML-SOR [95], ONSIDE [96], and SCORP [97] are examples of social context-based routing protocols. Other proposals, such as DFA [98], follow a hybrid approach that aims at leveraging the favorable properties of multiple domains, including mobility, probabilistic, and social aspects of nodes. We refer the reader to the cited references for a detailed description of each solution.

Context-oblivious protocols have the advantage of simplicity - they do not need any knowledge about the environment or the neighboring nodes, and they can guarantee low packet delivery latency at the cost of high bandwidth and memory overhead. Typical approaches to reduce resource consumption include using a Time-To-Live (TTL) field in packets and limiting the

number of copies of a message that can be redistributed into the network (ER uses TTL, while SnW uses both techniques). Mobility-based routing protocols try to reduce the overhead of context-oblivious protocols by exploiting historical data on past encounters with other nodes or the knowledge about other nodes' mobility patterns. Finally, social context-based routing protocols are based on the observation that the nodes' movements mostly depend on their owners' habits, and so they extend the concept of mobility-based routing protocols by taking into account users' social relationships to predict nodes' mobility patterns [81].

However, while context-based routing approaches are capable of achieving a better resource efficiency compared to context-oblivious ones [51] [56], they also present several important shortcomings. First of all, routing protocols that leverage contextual information, e.g., about node mobility or social interactions between users, need to collect sufficient data before they can extract any useful knowledge from them. In turn, this requires devices to set aside a certain amount of resources to store and process contextual information (contact times, interests, etc.) about each node encountered in the past. The need for a data collection phase, whose duration can be neither overlooked nor easily predicted, and the resources required to leverage social and mobility information about the multitude of nodes and people that cross their paths in urban environments make the adoption of context-based protocols in these highly dynamic scenarios particularly challenging. Finally, context-based routing algorithms are usually designed to leverage contextual knowledge to find the best path to reach a single destination, and it would be hard to use them effectively to deliver data to multiple requestors in the same network [57].

These claims are also supported by the results presented in [58], in which Barzan et al. use mobility data obtained from real traces of mass events to show how flooding-based protocols can perform better or show comparable results to more complex routing solutions, such as PRoPHET. Moreover, [59] further backs up these observations by showing that the most complex routing solutions were designed to maintain end-to-end connectivity between nodes in mobile ad hoc networks (MANETs) and opportunistic networks, but their properties becomes superfluous, and sometimes even detrimental, in ICN, where no concept of end-to-end connection exists.

Finally, it is important to notice that almost the entirety of the research done in the fields of routing in DTNs, MANETs, and opportunistic networking typically assumes that nodes in the network are willing to share (a part of) their resources to deliver other nodes' messages [60]. This assumption is motivated partially by the need to simplify the system model, and partially by the consideration that, in many specific contexts, nodes are deployed and configured by the same party, and hence the possibility of uncooperative behaviors could be ruled out. However, this last case does not apply to next-generation scenarios, where nodes include individually owned smart devices and vehicles. Nonetheless, nodes in the network need to cooperate and invest significant amounts of resources to ensure data delivery, regardless of the type of routing solution used (context-aware routing schemes need to build up a thorough knowledge of the network in order to enable smart routing decisions, while context-oblivious schemes aim at flooding the network with multiple copies of the same message to increase the probability of one copy reaching its destination). Therefore, it is essential to devise a compelling resource

management system that can control resource sharing at the network level. The system should give applications the tools to influence the resource sharing process on the node, by specifying the type of information they need. At the same time, a fair and secure resource management system should also prevent malicious applications that can take advantage of cooperating nodes without contributing to the routing effort with some of the resources available on their node.

3.2. Caching

A lot of work has been done recently on ICN caching [8] [35] [61] [62] [63] [64] [65]. Caching techniques in ICN can be classified as *on-path*, where nodes can exploit only the information of content cached along the path from the source(s) to destination, and *off-path* strategies, where nodes can also exploit the information of content cached on nodes that do not belong to such path [35]. Implementations vary further according to the dimensioning, the sharing mechanisms, the decision policy (what content is placed at which caching node), the eviction policy [8], and the content popularity model used [62] [66]. Regardless of the different implementation choices, all approaches assume the availability of large, fast storage memory at all router nodes in order to implement efficient routing decisions and to enable opportunistic content caching [7] [8] [67]. None of these assumptions hold in next-generation communications scenarios.

Depending on the caching decision policy and the nature of the content, all, some, or none of the router nodes that take part in data forwarding will also cache (part of) the forwarded IOs (on-path caching). Additionally, implementations of off-path caching policies have nodes notify the other nodes in the network, or a centralized content object registry service, about the IOs they have in their cache to further improve the routing of interest messages [68]. However, off-path caching is more expensive than its on-path counterpart and, especially in presence of dynamic network conditions and unreliable links, its benefits might not be enough to balance out the increased demand on network resources.

A case could be made for in-network caching in wireless environments, thanks to its particularly favorable properties in presence of unreliable links and unstable paths [59], and in heterogeneous networks with MANET and cellular communications. For those scenarios, the possibility of retrieving cached content directly from end nodes could significantly improve the efficiency of routing and alleviate congestion on the cellular network [69]. The consequence of ubiquitous in-network caching is that the competition for one node's memory resources will not only occur among the multiple applications running on that node [1], but also with IOs that have been opportunistically cached for the purposes of improving the overall network performance, making cache sharing a more complex matter.

Cache sharing is an aspect of ICN that so far has received a relatively little attention. This is partially because of the large number of open challenges [37], but also because cache sharing does not represent a critical aspect in the wired Internet, which is the working environment targeted by most research studies on ICN. One of the few studies that focus on this topic is [8]. While most of the observations made rest upon the wired Internet environment, those

concerning the optimization of the cache sharing process still stand when taken to the next-generation scenario. The authors point out that the optimization of cache sharing depends on the contextual point of view, which can be either that of the users, the contents, or the providers. Additionally, they claim that there are basically two ways to manage memory sharing: using either a *fixed* (where each class of the competing applications receives a fixed share of the available memory) or a *dynamic* cache partitioning system (where dynamic sharing policies, such as priority-based, determine the amount of memory reserved for each application class).

We believe that research on cache sharing mechanisms is essential in next-generation scenarios, thanks to its potential to reduce bandwidth usage and latency in a scenario where most content requests originate in wireless networks composed of resource-constrained nodes. However, more effort on this topic is required before ICN solutions can support next-generation mobile applications. Also to this end, the techniques developed by research in the field of opportunistic networking present particularly attractive directions to investigate [70].

3.3. Frameworks

The literature already has a few examples of complete frameworks based on the ICN paradigm to support application communications in wireless and heterogeneous networking environments.

Amadeo et al. propose an ICN architecture for IEEE 802.11 MANETs called CHANET [71]. CHANET relies on naming to identify the content and it makes use of broadcast for the transmission of both interest packets and data. Other interesting techniques implemented in CHANET to increase effectiveness in the wireless environment are the overhearing of nearby nodes' transmissions and the local decision-making processes regarding packet forwarding. The results achieved experimentally show that their approach performs better than traditional TCP/IP-based solutions in terms of network overhead and download time. Despite the very interesting outcome, the authors do not investigate performance in large-scale, realistic scenarios, nor do they address critical topics such as resource consumption on the mobile nodes. Finally, the paper does not describe the nodes' mobility model used in their experiments.

The authors of [29] propose an opportunistic, content-centric architecture that takes advantage of the increasing number of pervasive systems available today to share content. The proposed solution, called the Information and Context Oriented Networking (ICON) framework, encompasses techniques that come from both the research fields of data-centric networking and opportunistic networking. ICON exploits caching strategies developed in ICN to share and place contents across devices in the network, and it relies on opportunistic strategies based on social- and location-based information, and the knowledge about application data usage, to route and forward the content to nodes interested in it. The authors describe the framework architecture and give details on the structure of exchanged messages. Finally, they discuss the applicability and feasibility of ICON and present the results obtained comparing their solution against other social-aware and social-oblivious frameworks based on opportunistic networking using the ONE simulator and a scenario based on real mobility traces. However, the decision-making process designed in ICON does not take into account the resources available on the

other devices, and so it cannot discover the presence of resource rich nodes in the network that might be exploited to increase the system performance. Additionally, the results presented are obtained in a simulated scenario where a single source node generates messages for one or more recipients that do not generate any messages; such scenarios fail to recreate the many-to-many communications scheme that characterizes next-generation communications environments.

4. Taking ICN to Next-generation Communications Scenarios

The ICN paradigm has several characteristics that make a compelling argument towards its adoption in the next-generation scenario. However, ICN was devised for very different operating conditions and, as the literature review in the previous Section pointed out, it is important to realize that some of the mechanisms proposed to implement its core functions cannot be reused as they are in the new scenario, and so the design of new techniques is required.

One such mechanism concerns the dissemination of information in the network. The design of the dissemination algorithm represents an aspect of critical importance, as it directly affects caching decisions (which nodes will store which IOs?), resource utilization (how much bandwidth is available for content dissemination?), and the capability of the network to deliver IOs to destinations. Therefore, dissemination algorithms have a significant impact on the system's performance and scalability as perceived by the users.

Most ICN proposals assume operation within a fixed network topology with resource-rich caching nodes and deal with IOs that are essentially static and long-lived content that should be preserved as long as possible. As a result, those proposals focus on *finding the best location for IOs* within the distributed caching infrastructure through coordinated or uncoordinated techniques. That assumption is consistent with a vision focused on mostly-wired Internet communications, but we expect that the exchange of information in next-generation scenarios will reflect much more what happens in reality in the physical world. Consequently, IOs will have highly variable lifetimes, their content will be dynamic and will depend on the human activities in the environment, and their scope might be limited geographically. ICN implementations for next-generation environments should thus focus on *finding the best caching node(s) for each IO*, considering the limited lifetime of IOs and attempting to aggressively exploit "ferry" caching nodes that could operate as bridges between disconnected portions of the network.

The presence of a high number of mobile devices, with limited resources and frequent disconnections from the network due to mobility and users' actions require smart communications and caching systems that can adapt to the resources available on each host and to the current network conditions. Such systems could exploit content awareness information, users' preferences, their social circles, the data they are interested in, as well as those of their neighbors and their friends, to tailor the most effective IO dissemination and caching strategy for each host. The design of resource-efficient communications and caching algorithms would enable a smart dissemination of IOs in challenged environments, thereby

making ICN a very promising paradigm to support mobile applications in next-generation scenarios.

To effectively reduce the gap between ICN and the requirements of next-generation scenarios, we need to devise solutions that fit the new operating conditions by taking into account the characteristics of the IOs exchanged and the communication patterns of modern and future applications. In fact, besides the technology and social facets, which we have already discussed in the previous Sections, the characteristics of the information generated and consumed by users and applications in the next-generation present many important and peculiar aspects that are essential to take into account. Outlining and analyzing those attributes is a first, essential step towards our final goal.

4.1. A New Information Model

Next-generation mobile applications offer services that interact with people and things, enriched by the processing of large quantities of data gathered from the many types of sensors that monitor the environment around the users. This strong link between users, things, and the surrounding environment considerably affects the type of content that is produced, which presents a few critical characteristics.

One such characteristic is the volatility of its value: with time, users tend to access content less and less, a fact that reflects the *obsolescent nature of information*. Modern social networks, where people are most interested in the newest pieces of content published by their friends and family, offer a great example of this aspect. Nonetheless, others services exhibit the same property: news, live event streaming, hazard/disaster alert, street traffic conditions, video-chat, etc. all rely on the quick distribution of new content, whose value (in the form of utility for the users) decreases quickly with time. Furthermore, some applications will only be interested in delivering the most recent data to the users. In such cases, the importance of data critically drops down as soon as more updated data becomes available. These characteristics call for the prompt identification of outdated IOs in the network, so that its resources can be managed properly to offer a better Quality of Experience (QoE) to the users.

Another property of the information is its *priority* [24]. Many applications produce IOs with different priority levels. For instance, smart city applications that support mobility of people might want to disseminate IOs containing traffic jams or storm alerts as quickly and as broadly as possible, even at the cost of preempting the dissemination of lower priority IOs. Applications might also want to explore layered coding techniques to improve the QoE offered to their users, thereby assigning higher priorities to the IOs containing reference information and lower priorities to the ones containing differential data, even if those IOs conceptually belong to the same service. As an example, weather forecast applications usually provide a complete set of related information, such as temperature forecasts, rainfall, weather and barometric pressure maps, etc., but most of the time users will only be interested in the weather forecast for their current location in the next couple of days. Similarly, live event streaming applications can provide multiple resolution levels for their video stream flows; basic frames will be transmitted in

higher priority IOs and delivered to all viewers, while higher resolution frames will be packed within lower priority IOs and disseminated in a best effort fashion.

When used properly, prioritization functions allow applications to give nodes in the network suggestions on how to cache, disseminate, and route chunks of the published IOs [72]. In general, higher priority messages should have access to a greater number of resources when compared to messages marked with a lower priority level. However, it would become very difficult for an application to decide on the priority level of produced IOs compared to those generated by other applications. In fact, the task of choosing the appropriate priority level should take into consideration several factors that are most likely unknown to applications and developers. Such factors include things like how important some service is for the user, given what he or she is doing or needs to do, what job the user does and if the services provided by the application are relevant to his or her job, if the provided service is enabling the user to interact with other persons or devices and how important to the user is the outcome of that interaction, and so on. Furthermore, decisions about the priority level to assign to some IO should also consider the priority and content of other IOs (or at least any information made available by metadata) currently in queue for transmission.

For this reason, we argue that the *concept of priority should be application-specific*, thus affecting how resources are allocated for different IOs produced by the same application, instead of the amount of resources to which each application has access. In fact, applications should have all the necessary knowledge to establish the most appropriate priority levels for IOs they generate. Application-specific priorities also prevent greedy applications from undesirably raising the priority level of their IOs to have access to larger amounts of bandwidth, memory, and processing power at the expenses of other applications.

Note that researchers have recently started to explore dynamic and recipient-specific priorities for IO dissemination, by exploring Value of Information (Vol) based concepts [73]. Vol represents a very interesting and compelling opportunity to optimize communications in next-generation scenarios and would integrate nicely in ICN middleware for next-generation communications. However, exploring the integration of ICN- and Vol-based approaches in next-generation environments lies outside the scope of the present paper and is left for future work.

4.2. Patterns and Models of communication

Next-generation services will present a communication model that significantly differs from traditional client-server applications. In fact, users will take on both the roles of content producers and content consumers, switching naturally from one to the other, or even playing both roles at the same time, in a very dynamic way. The role assumed, as well as the content's recipient(s), will depend on the applications running on one node, the content's nature, and the users' social context, current location, and habits. These aspects outline a many-to-many communication model, as opposed to the one-to-one communication model of client-server applications [74].

In many cases, next-generation services and applications will not be interested in specific IOs, but more generally in any updated information concerning one or more themes. For instance, future traffic and emergency notification applications will be interested in receiving any IO carrying updated data about the traffic conditions or news about present hazards, instead of only those that contain information about specific hazards or the traffic conditions of specific portions of the map. Similarly, future social applications that allow people to rate and comment about their experiences in restaurants, bars, and other places will be interested in retrieving any IO with updated ratings or new comments on nearby places, instead of only the IOs that carry the data about the places specifically requested by the users. This different pattern of communication in next-generation scenarios has two important consequences: first, it fosters nodes to receive and cache data that might become useful for other nodes in the next future, thus moving information delivery in the network towards a more distributed and collective effort and putting to good use the unused storage residing at the edge of the network [28] [75]; and second, it potentially reduces latency times when users request information that has already been (partially) stored on the node.

The ICN paradigm will adapt well to the communications patterns of next-generation scenarios, as it allows users to be content producers by publishing some content, and content consumers by notifying their interest in certain IOs and requesting their delivery. ICN supports many-to-many communications by allowing users to subscribe to multiple types of content and retrieve IOs from multiple sources at the same time [76] [77].

Nonetheless, most ICN implementations use a pull-based content delivery model, distinguish only between on- and off-path caching [35], and implement simple cache eviction policies, such as Least Recently Used (LRU) or Least Frequently Used (LFU) [33] [68], or probabilistic IO replacement policies [67]. Therefore, the context information about the type of content that nodes are interested in does not get considered for IO caching or routing/forwarding. The main reason is that researchers designed the ICN paradigm for the Internet assuming a wired, mostly static network infrastructure at the basis. However, in dynamic scenarios where nodes move frequently and fast, and where they are potentially interested in any IOs that carry information on certain themes, the pull model typically implemented by ICN solutions needlessly increases delivery latency, thereby reducing the length of connectivity windows between nodes and harming the efficacy of IO dissemination [78] [79].

Instead, studies and experiments proved push-based models to be more effective under highly dynamic conditions [2] [80] or when one-to-many and many-to-many communication schemes are involved [5]. Thus, the system would benefit from a model where interest messages do not need to get routed towards the closest content provider available to issue a request for a specific IO, but instead nodes use them to notify their neighbors about the IOs they need. After that, neighboring nodes are left to decide for the forwarding of each single IO they have in their cache. This shifts the push model in the direction of a more hybrid one, according to which nodes forward cached IOs towards their neighbors based on the interest messages they received from them. In addition, IO caching techniques could also leverage knowledge about nodes' interests to improve their effectiveness.

5. ICeDiM: an Information-Centric Dissemination Middleware

We propose the Information-Centric Dissemination Middleware, or ICeDiM, to address the need for an effective ICN-based solution in next-generation communications scenarios. ICeDiM offers overlying applications the necessary set of services to enable the delivery of IOs in ICN-based heterogeneous networks. ICeDiM addresses the problems of resource sharing in next-generation networking scenarios, which include engaging people in sharing their devices' resources and the management of shared resources, by introducing the concept of Application-level Dissemination Channels (ADCs) as a unit of resource allocation and communication. In addition, ICeDiM leverages modified versions of the well-known Epidemic and Spray and Wait routing algorithms that take advantage of broadcast communications to improve the performance of IO dissemination. Thanks to these features, ADC-based IO dissemination extends naturally to heterogeneous network technologies and environments. Applications can tune ADC control parameters, including IO forwarding order and cache eviction policies, to optimize resource usage and important network performance metrics such as message delivery ratio and latency. Finally, ICeDiM provides policies for deciding the forwarding order of IOs waiting for transmission and for controlling caching eviction that are completely independent from each other. All these functionalities contribute to the composition of a system that enables the distributed smart management of available resources for communication purposes.

5.1. Application-level Dissemination Channels

Application-level Dissemination Channels (ADCs) are the fundamental building block for ICeDiM communications: they represent both an application-level thematic attribute and a node-level commitment to sharing resources within that channel. Thus, ADCs regulate resource consumption and nodes' collaboration in the effort of delivering messages in next-generation networks. Within the resource constraints introduced by ADCs, ICeDiM leaves all major decisions concerning the routing and forwarding of IOs to the dissemination strategy, as discussed in Section 5.3. Examples of themes for which ADCs could be used include textual news, video stream news, social networking, wiki, smart city access, information from nearby sensors, etc. For the scope of our research, we assume that all nodes know about and have the permissions to communicate over all the existing ADCs in the network.

The thematic nature of ADCs allows grouping of similar applications and services to achieve three objectives: reducing resource usage at the network level, overcoming nodes' and users' reluctance in sharing their resources, and exploiting potential optimization opportunities. Since only nodes interested in a given theme will participate to the corresponding ADC, ICeDiM keeps resource consumption, e.g., for IO retransmissions and storage and for processing of IOs and metadata, at a minimum.

Applications can join one or more ADCs dynamically, either by publishing IOs within that (those) channel(s), or by explicitly subscribing to IOs published within that (those) channel(s). When an

application joins some ADC, ICeDiM ensures that a part of the node's network bandwidth, storage capacity, and computational power is traded in exchange for support from the message delivery process from other nodes that subscribed to the same ADC. This way, all nodes belonging to some ADC will take part in the common effort of distributing IOs published within that channel, sharing their resources (storage space, contact opportunities with other nodes as they come in communication range, etc.). Thus, in the highly dynamic next-generation environment, each ADC effectively forms an impromptu and dynamic overlay network that applications can leverage to communicate.

We expect that applications and users will be more willing to share the resources of their devices if used to sustain communications for the same services to which they subscribed. Finally, if nodes cache IOs produced by applications of the same thematic classes, then there is a higher chance that future requests can be resolved directly from the cache.

ADCs are not restricted to a single application, so various applications at the same time can access the same ADC. When this happens, applications members of the same ADC will share resources. As a result, if the applications running on a node subscribed to multiple themes, that node would also participate in all the corresponding ADCs. Unsubscribing from the messages published within some ADC is equivalent to leaving that channel.

Finally, the thematic nature of ADCs allows application designers to explore trade-offs between the effectiveness of communications within the subscribed ADCs and the preservation of nodes' resources, by modulating the granularity of the dissemination channels' themes – and hence the corresponding number of participating entities. Building applications on top of a large number of ADCs with finer granularity themes would lead to a lower resource consumption on the nodes, but also to a lower collaboration between them and, consequently, diminished support from the network. In contrast, applications using a low number of ADCs with coarser granularity themes would limit the constraints enforced on resource consumption, to the point where all IO forwarding and caching decisions would be left entirely to the dissemination strategy (when a single ADC is used for all applications).

To illustrate how ADCs work, Fig. 2 gives a visual representation of the overlay networks generated by three different ADCs in a mobile scenario, represented with three different colors. We will name those channels as “Green”, “Yellow”, and “Violet”, respectively, from the colors used to distinguish them in the figure². Nodes A, B, and C are subscribed to “Green” ADC, nodes D and E are subscribed to “Yellow” ADC, and nodes G, H, I, and J are subscribed to “Violet” ADC. However, node A also subscribed to “Violet” ADC and, similarly, nodes E and F also subscribed to “Green” ADC. Finally, note that, in this example, node K did not subscribe to any channel. This might happen for several reasons, such as in cases where no applications are running on one node, or under request of the user, for instance in order to save the

² For performance reasons, ICeDiM uses a channel identification number, also referred to as ADC ID, to identify each ADC univocally. However, for clarity purposes, we will often use string representations to identify ADCs in our examples.

remaining battery life on his smartphone.

Black, continuous lines in Fig. 2 represent links between nodes that belong to the same ADC. ICeDiM can thereby exploit such links to transfer IOs within the context of that ADC. Transfer will take place as long as both the destination and the IO belong to the same ADC and the decision-making component within ICeDiM decides to begin transmission, in accordance with the underlying dissemination strategy. We call this type of transmission “*within-channel transmission*”. Instead, red dotted lines represent links between nodes that do not belong to the same ADC. Therefore, even though some connections are available at the link layer, ICeDiM ignores them for the purpose of IO forwarding. This way, for example, communications between nodes A and F are possible only by means of the store-carry-forward paradigm, i.e. by caching IOs and waiting for the two nodes to move under each other’s range.

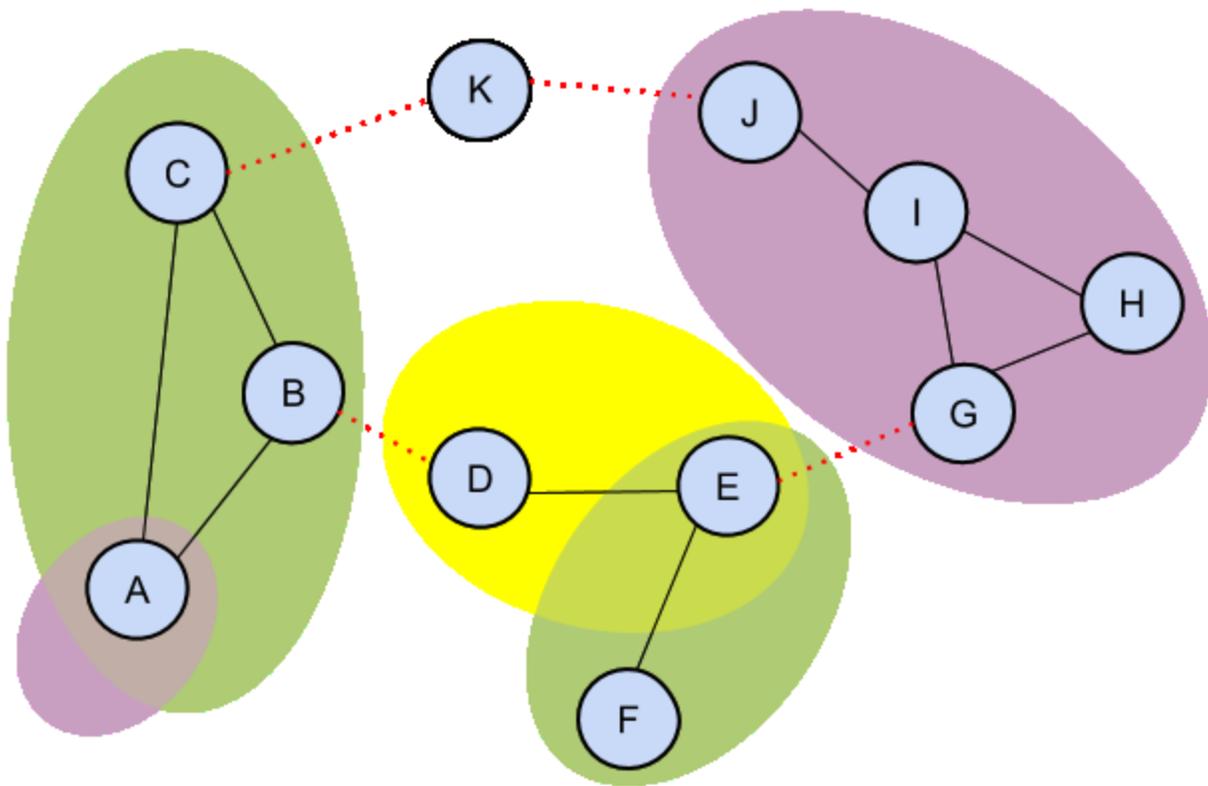


Fig. 2. ICeDiM nodes communicating over (Strict) Application-level Dissemination Channels.

5.2. Permeability of Dissemination Channels

Many different types of nodes, owned by various parties (public organizations, private companies, individual people, etc.), with different and continuously changing goals and constraints, will operate in next-generation networking environments. For instance, the type of network activity performed by personal devices such as smartphones, tablets, and laptops highly depends on the user behavior, as well as the screen size and resolution of their devices,

remaining battery life, computational power, available memory, and so on.

In these situations, assuming that all nodes will always cooperate with the IO dissemination process would be too optimistic. As a result, strictly enforcing the dissemination of IOs within the corresponding ADC boundaries might lead to missed connectivity opportunities, prevent message delivery to interested parties, and cause partitioning phenomena in the overlay networks associated to ADCs, even if an end-to-end path between two nodes is available. This is illustrated clearly in Fig. 2, which shows two different network partitions for the “Green” ADC, corresponding to nodes A, B, and C and nodes E and F respectively.

To address these issues, ICeDiM implements a mechanism that enables the partial relaxation of the constraints on IO dissemination imposed by ADCs. To this end, borrowing from the metaphor of molecule diffusion processes in biological systems, where a portion of the molecules contained in a substance can cross a cell’s membrane, we introduce the concept of *dissemination channels with permeable boundaries*. This mechanism allows a (configurable) portion of IOs to pass through the boundaries of an ADC to favor their dissemination. The goal is to maintain nodes’ resource consumption and participation in the IO dissemination process under control, while increasing the options for communication.

More specifically, ICeDiM supports three types of ADCs (or ADC modes), with different permeability levels: *Strict*, *Semipermeable* (a term commonly used in biology to describe the behavior of cell membranes), and *Unconstrained*. The level of permeability adopted for an ADC is a fundamental and system-wide attribute, set by ICeDiM at the moment of the ADC creation. From that moment on, all nodes subscribing to that ADC must comply with the IO dissemination rules defined by its permeability level.

Strict ADC (S-ADCs) have impassable boundaries, as described in Section 5.1. As a result, for an S-ADC, ICeDiM nodes only allow within-channel IO transmissions and cache IOs that belong to one of the thematic channels they have joined. Hence, a node can disseminate an IO “M” over an S-ADC “C” to other nodes only if they subscribed to “C” as well.

Semipermeable ADCs (SP-ADCs) permit a fraction of IOs to seep through the boundaries of their own channels and, thus, reach portions of the network that would be inaccessible using S-ADCs. Thereby, in addition to within-channel transmission of IOs, semipermeable channels also implement a diffusion process that comes into play to determine if an IO can pass through the “channel membrane”, thus becoming eligible for transmission to nodes that did not subscribe to its ADC. Similarly, ICeDiM nodes might cache IOs received over an SP-ADC that does not belong to any of the ADCs joined by the nodes. In any case, within-channel transmissions have higher priority than those generated by diffusion, i.e., ICeDiM favors the transfer of IOs that can be delivered to nodes in the same ADC over transfer of IOs that do not belong to any of the ADC that neighbors have joined. In the remainder of this paper, we will refer to the transmission of an IO across the boundaries of its ADC, as enabled by diffusion, as a *“cross-channel transmission”*.

Finally, in *Unconstrained ADC* (U-ADC) no boundaries whatsoever are enforced on the dissemination of IOs. This represents a degenerate case, in which all nodes in the network have joined the same, single dissemination channel on which applications may send any IO, independently from its corresponding thematic attribute. The U-ADC mode does not enforce any constraint on resource allocation and consumption in the nodes; thereby, IO forwarding and caching decisions are left entirely to the dissemination strategy.

Fig. 3 shows the same scenario of Fig. 2, but, in this case, semipermeable ADCs are used instead of strict ones. The main difference is the use of black dashed lines to represent links between nodes that do not belong to the same dissemination channel, instead of red dotted lines, to show that IO transmission across different ADCs is permitted when the SP-ADC mode is used. Let us consider node A in Fig. 3, which hosts an application that has published an IO within the context of the “Green” ADC (depicted as a green unopened letter in the figure). ICeDiM transfers the IO to B by means of a within-channel transmission, since A and B both belong to the “Green” ADC. At this point, B can try to transfer that IO to D, which might decide to cache it, even if not interested in “Green IOs”. If this happens, then diffusion was successful and the IO has effectively passed through the channel’s membrane to be received and stored by node D. At this point, D knows that one of its neighbors, namely node E, has subscribed to the “Green” ADC, and so it will deliver the IO to E by means of a within-channel transmission. After reception is complete, node E can transfer the IO to node F as well, by performing a third within-channel transmission. Note that the transmission between nodes D and E is within-channel because both the IO and E belong to the “Green” ADC, and not because nodes D and E both joined the “Yellow” ADC.

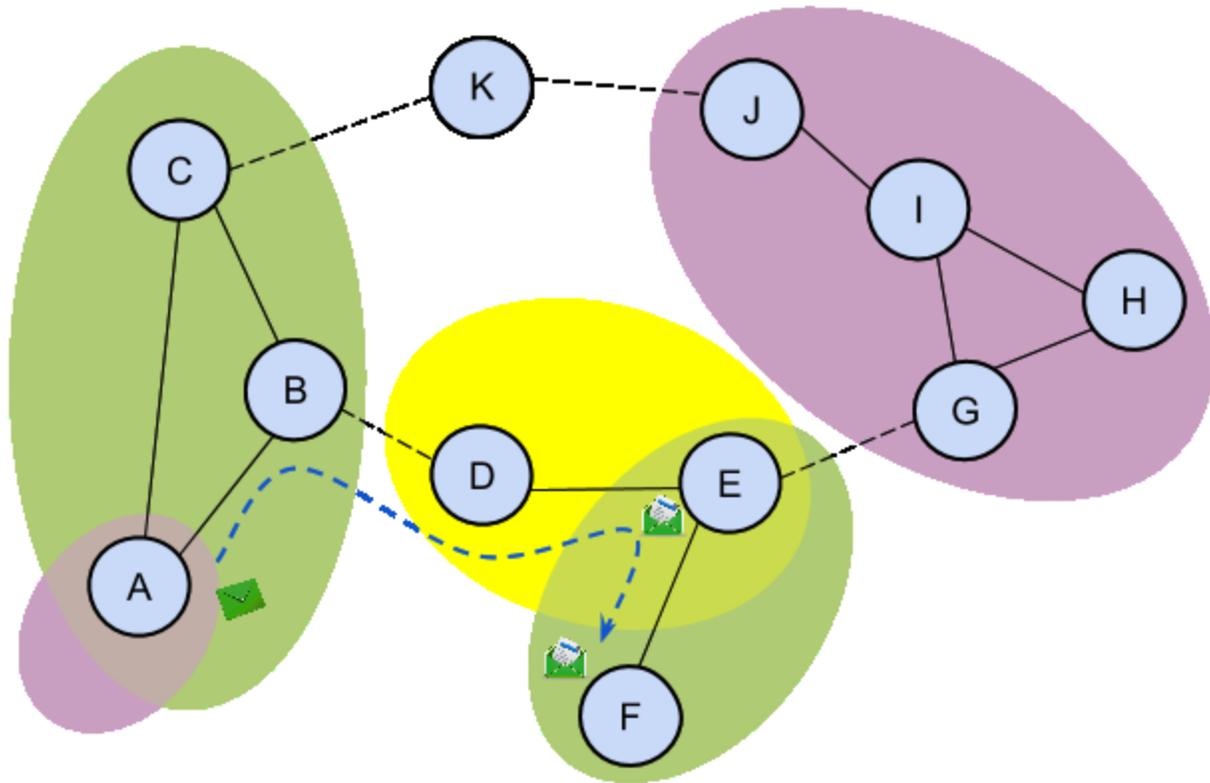


Fig. 3. ICDiM nodes communicating with Semipermeable Application-level Dissemination Channels.

5.2.1. Controlling the Permeability of Dissemination Channels

ICDiM uses a model based on two random procedures, controlled by different (and fixed) probability parameters, to regulate the diffusion or “membrane-passing” phenomenon, i.e., the process of one IO seeping through the boundaries of its SP-ADC and (potentially) being received and cached by another nearby node that did not join the same channel.

Again, this behavior is inspired to biological systems, where the membrane-passing diffusion process has an intrinsically random nature, as not all molecules will cross a cell’s boundaries. However, in biological systems, the intensity of the diffusion process is proportional to the concentration gradient of the molecules across the two sides of the membrane: the higher the concentration gradient, the higher the probability of molecules to seep through the boundary.

Instead, ICDiM needs a distributed mechanism that regulates the intensity of the diffusion process independently from the amount of IOs exchanged within an SP-ADC (and within the other ADCs), as well as to control the forwarding and caching aspects of the IO diffusion process in an independent fashion. To this end, each ICDiM node independently contributes to the implementation of the IO diffusion process within an SP-ADC by running two forwarding and caching decision-making procedures, which have a random behavior controlled, respectively, by the Forwarding Probability (FP) and Caching Probability (CP) parameters set at the SP-ADC

level.

Thus, whenever a node has the possibility to perform a cross-channel IO transmission, ICeDiM will send the IO with probability FP. Similarly, upon reception of a new IO following a neighbor's cross-channel transmission, ICeDiM will cache the received IO with probability CP. Whenever a node generates a new IO or caches it for the first time, ICeDiM looks at the local database of ADCs subscribed by the node's neighbors, which, for convenience, we call "*Receivers Channel Set*" (RCS). If the IO's ADC does not appear in the RCS, ICeDiM will run the forwarding decision-making procedure before transmitting the IO. With reference to our biology-inspired metaphor, a successful diffusion process is equivalent to an IO transmission at the sender node followed by IO caching at the receiver node. Thereby, we identify two halves of the diffusion process in ICeDiM: a forwarding half, controlled by the FP parameter, and a receiving half, controlled by the CP parameter.

ICeDiM keeps track of all forwarding and caching decisions made by the local node concerning each IO until its expiration, and reevaluates them only when network conditions change in a way that could open new communication opportunities; this ensures that ICeDiM can discover and take advantage of those opportunities. Saving past decisions is a fundamental step for a correct implementation of the diffusion process and it allows ICeDiM to save network resources. In fact, naïvely reevaluating the forwarding and caching decision-making procedures every time ICeDiM needs to choose whether to forward or cache an IO would severely distort the impact of FP and CP, increasing their effective value of a factor that depends from the number of contact opportunities, the duration of those contacts, and the dissemination strategy employed.

To understand this problem, let us assume that nodes X and Y in Fig. 4a run a naïve implementation of the SP-ADC mode and that X has an IO "M" in its cache that does not belong to any of the ADCs joined by Y. In our example, Y has entered the connectivity range of X, which chooses to forward M to Y, following a successful outcome of the stochastic IO forwarding decision-making procedure controlled by FP (which implements the first half of the diffusion process). As shown in Fig. 4a, Y decides to drop M, following a negative outcome of the stochastic IO caching decision-making process controlled by CP, thus marking the diffusion process as unsuccessful. The two nodes then proceed towards their destinations and lose their connection. After a while, Y enters the connectivity range of a third node, Z, which also happens to have M in its cache (Fig. 4b). Similarly to what X did before, following a successful outcome of the stochastic IO forwarding decision-making procedure, Z forwards M to Y. However, as shown in Fig. 4b, this time Y decides to cache the message, following a re-run of the stochastic IO caching decision-making process that returned a successful outcome. As a result, M has

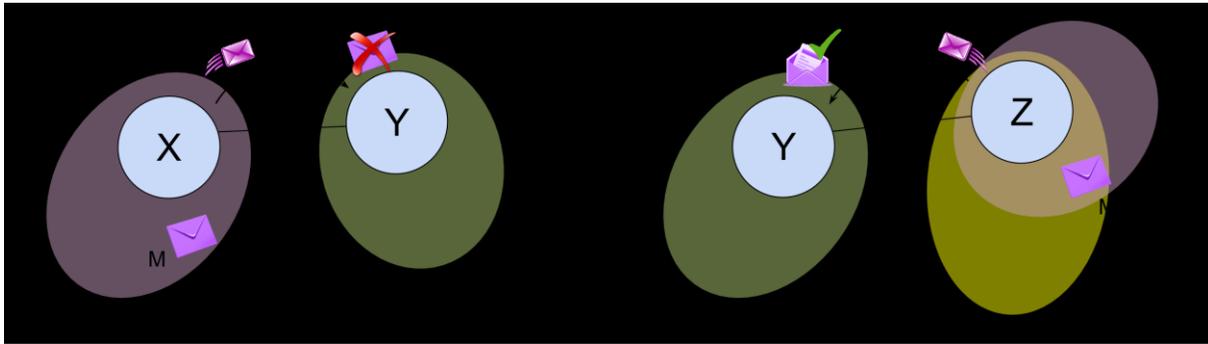


Fig. 4. Naïve implementation of the diffusion process: an unsuccessful diffusion of M (a) followed by a successful diffusion (b)

been forwarded twice to Y before the node decided to cache it, doubling bandwidth consumption and increasing latency.

This issue comes from a naïve implementation of the diffusion algorithm, which effectively reduces the control provided by the stochastic IO caching decision-making procedure. More specifically, this version of the algorithm causes receiver nodes to decide whether to cache IOs or not each time they receive one, even in the case of a copy of an IO that was already discarded in the past. As shown in the example above, this could lead to very significant performance losses in real-life scenarios, where large numbers of nodes and IOs are involved. Additionally, a selfish node could exploit this logic to force the dissemination of some IOs at the expenses of other IOs in the system, simply by retransmitting those IOs until the receiver nodes decide to cache them.

To tackle these problems, we adopted a more sophisticated decision-making process for both phases of the diffusion algorithm that reevaluates past decisions only when appropriate. More specifically, in ICeDiM, all forwarding-related decisions concerning an IO “M” (that is, whether to transmit it or not) that a node X makes are kept valid until a new neighbor Y that is not subscribed to M’s ADC enters the transmission range of X. Note that nodes leaving X’s transmission range are ignored, which means that neighbors’ disappearance does not trigger the reevaluation of any IO forwarding decision made by a node. ICeDiM implements this feature by keeping a “Forwarding Diffusion Table” (FDT) where entries record the result of the last forwarding decision made for each IO in cache (“forward” vs. “do not forward”). When a new neighbor enters one node’s transmission range, ICeDiM compares the RCS with the ADCs of the IOs in the FDT. For each IO whose ADC is not in the RCS, the corresponding entry in the FDT is cleared so that the forwarding decision can be made anew.

Differently, for the receiving phase, the decision of caching or discarding an IO is made only once upon IO reception and then nodes comply with it indefinitely. In order to do this, ICeDiM nodes keep an “Acceptance Diffusion Table” (ADT) where each entry indicates whether a specific IO has been marked as accepted or discarded. ICeDiM periodically cleans this table from entries referring to expired IOs, to avoid pointless memory consumption and keep the table look-up time low. We verified experimentally that this implementation of the diffusion algorithm,

combined with the adaptation of the anti-entropy phase of the underlying routing protocols (described in the next Section), prevents that nodes transmit the same IO more than once to the same receiver, thus saving significant network bandwidth and also reducing latency. Moreover, the updated version of the diffusion algorithm makes it impossible for selfish nodes to force the dissemination of certain IOs by repeatedly sending copies to nearby nodes, as the receivers' decision of caching or discarding an IO is final.

5.3. Dissemination Strategies for Wireless Communications

Wireless technologies such as Wi-Fi allow nodes to broadcast messages to all neighbors at the same time. In terms of bandwidth usage, the cost of broadcasting a packet is (almost) the same as sending the same packet to a single node via unicast, but broadcast can reach multiple nearby nodes in a single transmission, thereby potentially increasing spectrum efficiency [2]. This fact is widely known in literature and many techniques designed to improve network performances rely on it [82] [83].

The broadcast nature of the wireless medium represents a significant potential also for the ICN paradigm. In fact, while nodes on the delivery path of an IO will normally cache and forward the data towards the subscribers, other nearby nodes can exploit the broadcast transmission to know what IO is being delivered and any useful pieces of information that can be extracted from the metadata sent along with the IO. Additionally, broadcast transmissions can ease the delivery of IOs by enabling the opportunistic caching and redistribution of overheard objects to other nodes in the network. One could see this strategy as something in between on-path and off-path caching, and it is particularly effective in those cases where multiple nodes in one network have subscribed to the same content, since it enables the network to take a distributed, proactive effort to deliver the content to all subscribers [36].

This strategy can significantly reduce resource consumption when, for instance, many customers sitting in a coffee bar have connected their tablets and smartphones to the free Wi-Fi to watch a news channel that is streaming breaking news, or a live game of the national football team. In such cases, the AP can broadcast messages to the devices in the ICN-enabled network of the coffee bar, thereby saving resources and allowing nodes within range to cache transmitted messages. Live video streaming applications could implement features like, for example, a reliable, even if lower-quality, video stream, or the generation of a limited amount of retransmission requests to retrieve frames missing from its internal buffer, in order to increase the offered streaming resolution. The combination of the availability of frames cached by nodes of the local network and the content-centric delivery mechanism of ICN allows these and any applications with similar features to retrieve missing data from nearby nodes, consequently reducing server-side load. Moreover, resolving retransmission requests locally ensures more timely data retrieval, which is critical for services like live video streaming. Detti et al. propose and analyze a similar approach to increase efficiency of video streaming over ICN-enabled networks [84].

Nodes running ICeDiM can take advantage of broadcast-enabled network interfaces by

periodically sending HELLO messages to notify neighboring nodes about their presence. HELLO messages contain information about all the ADCs the sender has joined (said otherwise, HELLO messages contain all sender's interests) and all the IDs (each one being a string of variable length) of cached IOs. ICeDiM nodes use the information stored in the HELLO messages sent by neighbors to update the local RCS. During forwarding, ICeDiM uses the information contained in the RCS to select which IOs to send and passes them to the underlying dissemination strategy. IO selection is done by matching the ADC joined by neighbors with those of cached IOs. If an IO forwarding order prioritization strategy is present, ICeDiM applies it to the list of IOs available for forwarding before feeding it to the dissemination strategy. Otherwise, the forwarding order is determined by the cache eviction policy, which establishes the order of both IO eviction and IO extraction from cache.

Normally, ICeDiM caches all messages it receives that belong to one of the channels the node has joined or that were probabilistically accepted if SP-ADCs were used. In addition, ICeDiM can take advantage of connectivity options and features that are only available on a subset of the interfaces. For instance, if a node is equipped with a network interface that supports packet broadcast, ICeDiM applies an opportunistic caching strategy that permits caching overheard IOs broadcasted by neighbors. Note that the rules of ADC are still valid: caching will not occur if the IOs do not belong to one of the ADCs joined by the overhearing node or if they did not manage to traverse its semipermeable channels.

If the memory allocated for IO caching is full when ICeDiM receives a new IO, cache eviction occurs. The cache eviction algorithm implemented in ICeDiM removes one IO at a time until enough space is left to store the new IO. However, to avoid discarding higher priority IOs in favor of a less important one, the received IO also takes part into cache eviction. Finally, as we will discuss in more detail in Section 5.4, the delivery of newly received IOs to applications occurs before the attempt to cache them, so a full cache never jeopardizes IO delivery.

We included two broadcast-capable dissemination strategies in ICeDiM: the Epidemic Broadcast Router (EBR) and the Spray and Wait Broadcast Router (SnWBR). These strategies implement the broadcast versions of the Epidemic and the Spray and Wait routing protocols for MANETs, as documented in [49] and in [50]. The main difference introduced by the broadcast versions of the two strategies is the use of broadcast instead of unicast to transmit packets, whereas the logic that handles packet (re-)transmission and discard from memory is unchanged.

We also changed the versions of EBR and SnWBR implemented in ICeDiM to use the ADC ID in place of the address of the destination node. This way, ICeDiM can rely on the mechanisms implemented within EBR and SnWBR to deliver IOs in the context of ADCs. Note that this solution can be used in combination with any one of the three ADC modes, as both EBR and SnWBR can leverage neighboring nodes that are not a destination to disseminate IOs. However, depending on the ADC mode set and the current RCS, ICeDiM will pass to the underlying dissemination strategy only IOs whose ADC belongs to the RCS (if the S-ADC mode is selected), or also IOs whose ADC is not part of the channels currently in the RCS (if SP-ADC

or U-ADC are chosen).

Finally, we made one last change on EBR and SnWBR that involves the anti-entropy phase of the protocol. The original versions of the Epidemic and Spray and Wait routing protocols have an anti-entropy phase to exchange information about what messages each node has in its cache, so that nodes can learn which messages should be transmitted and to which node. We have extended the anti-entropy phase of both the EBR and SnWBR and integrated it in the ICeDiM's HELLO messages, so that nodes are capable of notifying their neighbors about the decision of not caching certain IOs, as it might happen when SP-ADCs are used. Combined with the second phase of ICeDiM's diffusion algorithm, the adaptation of the anti-entropy phase prevents ICeDiM nodes from performing pointless transmissions, thereby saving bandwidth and enabling a better utilization of the communication opportunities.

Another very interesting technology that can further increase efficiency in next-generation networks is Device-to-Device (D2D) communications. Some of the D2D techniques that have been proposed exploit the cellular spectrum to establish direct communications between two mobile devices without the support from base stations or the core of the cellular network. The advantages of this technology include higher throughput, better energy efficiency, lower communications delay, and increased fairness [16]. Several types of services could benefit greatly from it, e.g., video communication, cellular network offloading, supporting communications during disaster recovery, proximity-based services, etc. D2D is currently a very active research topic, but many problems still need a resolution and many design choices are yet to be made before devices will support this technology.

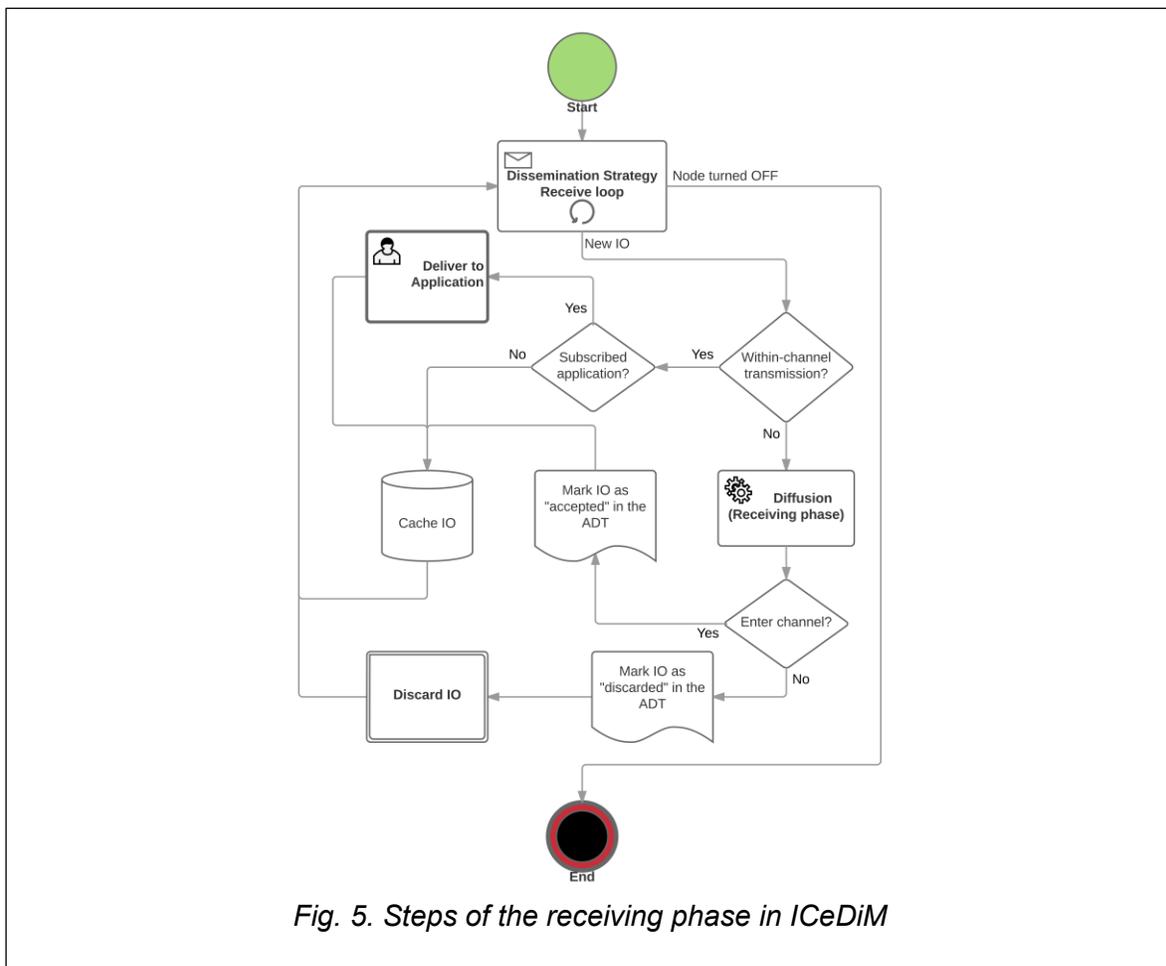
Nonetheless, ICN-based communications middleware such as ICeDiM would benefit greatly from D2D techniques. In fact, the availability of an additional network interface to perform local communications would broaden nodes' communication opportunities, thus improving the quality of routing and forwarding decisions. D2D would enrich the heterogeneity of connectivity technologies the middleware can use by adding one that provides one-to-one node communications characterized by low latency, low energy consumption, and high throughput. Studies show that the combination of D2D with other wireless short-range technologies, like Wi-Fi, can be extremely effective to reduce power consumption and service latency in multicast-based applications [85]. Additionally, D2D can be used to support content sharing among nearby nodes and extend the network coverage. These features are key blocks of ICN-based communications in next-generation environments, and so we believe that D2D will become an enabling technology for ICN in those scenarios.

5.4. Insights on ICeDiM IO Management

In order to illustrate how ICeDiM realizes ICN-based communications, we present an in-depth description of the IO management procedures it implements. For simplicity and without loss of generality (as the other types of ADCs are indeed a simpler case of SP-ADC), we focus on the IO receiving and forwarding phases for SP-ADCs, which we analyze separately to show how they contribute to ICeDiM's decision making in IO caching and forwarding. The two phases are

described by the block diagrams depicted in Fig. 5 and Fig. 6.

Fig. 5 shows all main steps that compose the receiving phase of an SP-ADC in ICeDiM. When a node is turned on, ICeDiM enters a receive loop that waits for new IOs to come in. When it receives a new IO M, ICeDiM checks if it was exchanged via a within-channel transmission, i.e., if the node is subscribed to the ADC to which M belongs. If so, two things can happen: either M needs to be delivered to one of the applications running on the node, or it only needs to be stored in memory. In the former case, ICeDiM takes care of delivering M to the application(s) by consulting an internal database it maintains to keep track of all interests notified by applications. Regardless of the fact that M has been delivered to the application, ICeDiM proceeds by caching M (not shown in the figure) and then reentering the receive loop, where it will wait for more IOs.



On the other hand, if M was exchanged via a cross-channel transmission, i.e., M does not belong to any of the ADC subscribed by the local node, then ICeDiM enters the receiving half of the diffusion process. In this phase, ICeDiM first checks the ADT to see if there is already an entry for M. If this is the case, ICeDiM simply applies the decision found in the ADT entry. Otherwise, ICeDiM runs the caching decision-making procedure with the value of CP set for M's SP-ADC, stores the value obtained in the ADT for future access, and follows up accordingly. If

the decision was to accept the IO, ICeDiM stores it in the cache; if, on the opposite, the decision was to discard the IO, ICeDiM drops it. In any case, immediately after caching or discarding the IO, ICeDiM goes back to the receive loop.

Fig. 6 illustrates the main steps involved in the forwarding phase of ICeDiM. At startup, the middleware checks the RCS to retrieve information on its neighbors. Updates to the local RCS database are triggered by the reception of HELLO messages (not shown in the diagram). Then, ICeDiM accesses the memory for a list of IOs that match the information retrieved at the previous step. If the list is empty, ICeDiM goes back to reading from the RCS; otherwise, the forwarding phase continues. If an IO forwarding order prioritization strategy is present, ICeDiM applies it to the retrieved list; if not, ICeDiM keeps the order of the list, as explained in the previous Section. Then, ICeDiM scans through the ordered list to look for an IO eligible for within-channel transmission, entrusts the first one it encounters to the dissemination strategy, and goes back to the beginning.

If no within-channel transmission is available, then ICeDiM enters the forwarding phase of the diffusion process. First, it checks if the FDT already contains an entry for the selected IO. If the entry is present, it means that ICeDiM already went through the diffusion process for that IO and, depending on its result, ICeDiM will either deliver the IO to the dissemination strategy or do nothing and go back to the beginning. On the other hand, if the FDT contains no entry for the selected IO, ICeDiM runs the forwarding decision-making procedure with the value of FP set for the IO's SP-ADC and stores the result of that decision in the FDT for future access. Next, if the decision was to forward the IO, ICeDiM dispatches it to the dissemination strategy. Finally, ICeDiM goes back to reading from the RCS, ready for another iteration. To avoid processing the same IOs in following iterations, ICeDiM implements a penalty mechanism that de-prioritizes IOs each time they are selected for forwarding.

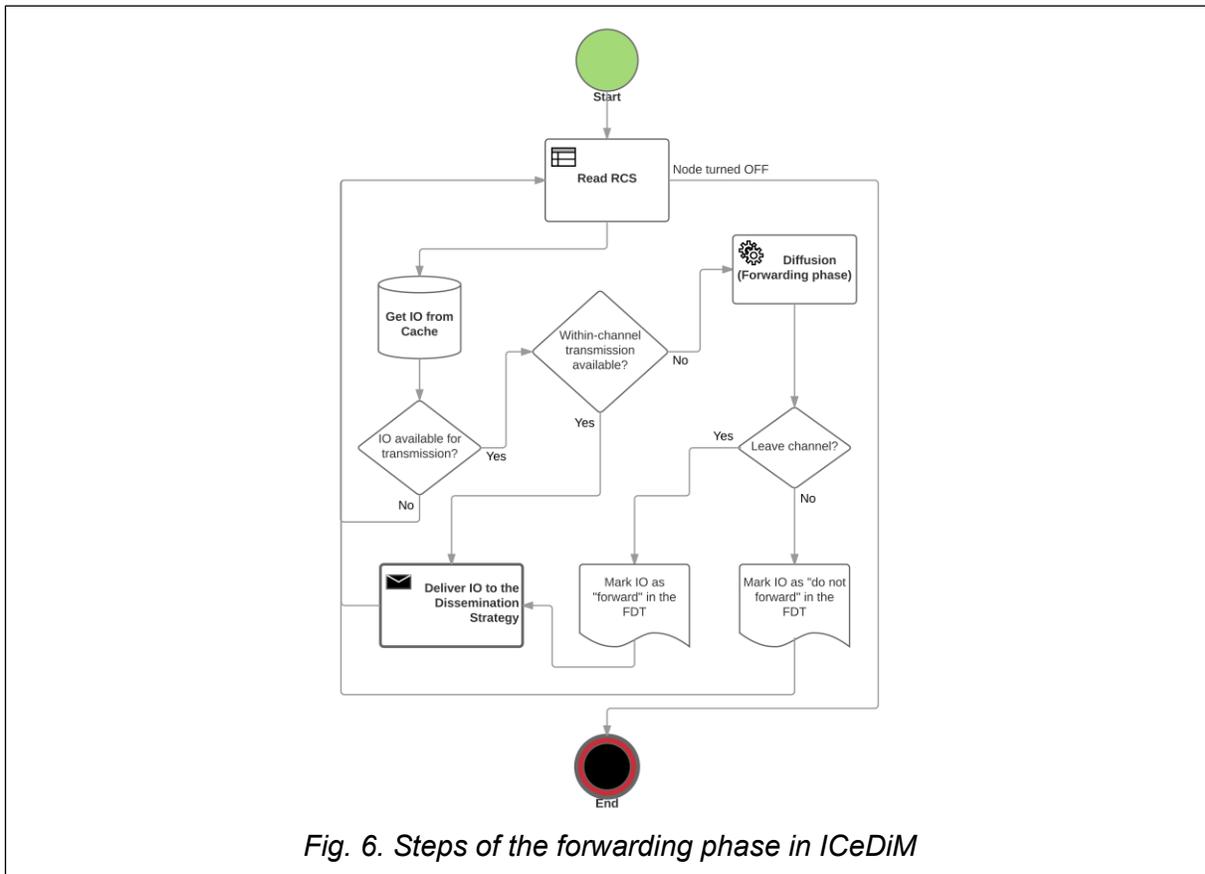


Fig. 6. Steps of the forwarding phase in ICeDiM

6. An improved version of the ONE Simulator

Given the difficulty of running large scale experiments in a real scenario with the number of systems and agents that would be needed, we opted to use simulation and reproduce the conditions of next-generation environments. The importance of simulation in the area of wireless and heterogeneous networks is widely recognized in the literature [58] [86] [87] [88]. We evaluated several different simulators before choosing the one that best fit our needs.

6.1. Choice of the Simulation Engine

NS-2 (<http://nslam.isi.edu/nslam/index.php>) and NS-3 (<http://www.nslam.org/>) provide very accurate and comprehensive simulation environments that supports many standard protocols and network interfaces. However, NS-2/NS-3 do not provide direct support for realistic mobility models, but require other tools to generate traces from such models to feed the simulator. Similarly, GTNets (<http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/>) provide an accurate network simulation environment that aims to reproduce the structure of actual networks, but the support for mobility models is poor. Furthermore, neither NS-2/NS-3 nor GTNets are specifically designed for wireless ad hoc networks or for information dissemination: this makes the set-up of a smart city scenario and the implementation of the required dissemination algorithms extremely time-consuming and error-prone processes.

Unlike the previous solutions, TraNS (<http://lca.epfl.ch/projects/trans/>) is a GUI tool that integrates real vehicle mobility traces coming from SUMO (a road traffic simulation package, <http://sumo.dlr.de>) with the network simulator NS-2, to allow the realistic simulation of Vehicular Ad hoc Networks (VANETs). Similarly, VanetMobiSim (<http://vanet.eurecom.fr/>) can generate movement traces of vehicles in different formats, thus supporting the integration of different network simulation/emulation tools. Nonetheless, both TraNS and VanetMobiSim focus exclusively on the simulation of VANETs, while smart city scenarios require a more diverse and complex combination of nodes' mobility models, as people and infrastructure are also valuable resources. In addition, besides the support for mobility models that can capture the heterogeneity of nodes in a smart city, the accurate analysis of the features of ICeDiM require a proper networking simulation engine.

After having considered the alternatives listed above, we chose to use a modified version of the ONE Simulator (<http://www.netlab.tkk.fi/tutkimus/dtn/theone/>) [89] for our experiments. ONE is open source, written in Java, and it is designed to simulate wireless opportunistic networks. These characteristics mainly drove our choice, as we were able to extend ONE with the missing features we needed to implement ICeDiM and to set up our experiments fairly easily. Another value of ONE is that it comes with a set of Router classes that implement a number of routing algorithms for MANETs: Epidemic, Spray and Wait, PRoPHET, MaxProp, and others. ONE supports several types of nodes (pedestrians, cars, trams, buses, etc.) and mobility models and allows both the pseudo-random generation of events, such as the generation of new messages, and the parsing of files containing event traces. In addition, setting up realistic smart city scenarios, using real maps such as that of the city of Helsinki, included in the simulator's bundle, is quite an easy task to do with ONE.

Notwithstanding the many useful characteristics of ONE, the accurate simulation of Application-level Dissemination Channels and broadcast wireless transmissions, as used in ICeDiM, require a few, yet very important, extensions. We built upon version 1.4.1 of ONE to introduce all the necessary missing features. We will hereinafter refer to our extended version of the ONE simulator as the ICeONE (Information-Centric ONE) simulator, or simply ICeONE. ICeONE is available for download at the link <http://endif.unife.it/dsg/iceone>.

6.2. Additional features of ICeONE

Arguably, the most important feature we added is the support for the Publish-Subscribe paradigm, necessary to implement ICN-based communications in ICeONE. In accordance with that paradigm, applications can publish messages in the context of a subscription, and they can subscribe to one or more subscriptions to receive messages published within them. In addition, nodes (and the applications running on them) can generate messages and mark them with a specific channel ID. As a result, marked messages will belong to the relative ADC and will have access to all the resources set aside for such a channel.

Another important improvement is the support for broadcast transmissions, which the original version of the simulator does not natively provide. We implemented broadcast transmissions in

the lower level components of the simulator. The motivation is twofold: to avoid polluting higher level components, which implement the dissemination policies, and to allow the design of other capabilities on top of the transmission mode, but independently from it. These functionalities include an extensible interface for the *management of packet collisions, which need to be accounted for to perform accurate simulations of wireless networking scenarios*, and a *collision avoidance mechanism* similar to the CSMA/CA mechanism described in the IEEE 802.11 specifications [90].

To manage packet collisions, the ICeONE Simulator requires any simulated network interface to be associated with an implementation of the Collision Model (CM); this allows configuration of different interfaces with the CM that better reproduces collisions for that particular technology. ICeONE already comes with two simple CM implementations, as well as a Java Interface that enables users of the simulator to develop their own models. In our experiments, we used what we called SPCM (Synchronized Packets Collision Model). SPCM permits the correct reception of packets only when the receiver remains within the sender's coverage during the entire transmission. In addition, SPCM marks any packet whose transfer overlapped with the reception or transmission of any another message, even partially, as a collision. Users could easily extend ICeONE with more sophisticated CMs that consider additional parameters when computing the probability of collision, such as the signal energy or the presence of obstacles in between. However, the accurate reproduction of each possible collision event is out of the scope of our study, which we limited to the cases that SPCM can handle.

The CSMA/CA algorithm we designed for ICeONE is based on the algorithm described in the IEEE 802.11 specifications. According to it, the wireless network interface senses the channel whenever the node needs to start a transmission. If the channel is sensed as busy, the transmission is backed off for a fixed amount of time after which the channel is sensed again, otherwise the node sends the entire message. CSMA/CA still leaves open the possibility of incurring the hidden or the exposed node problems [91], which can cause collisions or reduce network capacity. Note that while IEEE 802.11 provides for the use of collision avoidance before performing broadcast transmissions, the Request-to-Send/Clear-to-Send (RTS/CTS) mechanism is enabled only for unicast [90] [92], and ICeONE is compliant with the IEEE 802.11 standard.

7. Experimental Results

The following Section presents the sets of experiments we ran to assess the effectiveness of the concepts described in this paper in improving the process of information delivery. We used ICeONE to simulate a smart city scenario for our tests. We believe that the characteristics of smart cities, in terms of network heterogeneity and node mobility, provide one of the most compelling test cases for the evaluation of ICeDiM. We performed 3 different experiments, each one aimed at evaluating the impact of different features.

7.1. A smart city scenario

To evaluate the concepts and ideas presented in this paper in the most realistic conditions, we used the downtown Helsinki scenario included in the original ONE Simulator bundle. Nodes' movements are limited to streets. We differentiate walking nodes (pedestrians, identified by the letters "p" and "w"), car nodes (identified by the letter "c"), and tram nodes (identified by the letter "t"). 80 pedestrians walk along the streets with a random speed, uniformly chosen in the [0.5, 1.5] m/s range. Similarly, 40 cars move with a speed that ranges from 2.7 to 15.3 m/s, while 6 trams have a speed ranging from 7 to 11.1 m/s, for a total of 126 nodes. Pedestrians and cars choose a traveling speed and a random destination point in the map, reach that point through the shortest path, and then stop for a time randomly chosen according to a uniform distribution ranging from 0 to 120s. On the contrary, trams drive predefined routes back and forth, stopping for a time that ranges from 10 to 30s whenever they reach an end of the route, which simulates stops at a bus stop. Fig. 7 shows a screenshot of the nodes' positions at the beginning of the simulation.

Each node is equipped with a short-range, low-speed, and low-power wireless network interface, whose specifications are compatible with those of Bluetooth® 2.0 + EDR; trams also have a wireless interface comparable to IEEE 802.11 Wi-Fi installed on them, which allows for longer-range connections and higher bandwidth. Bluetooth-like interfaces have a coverage of 100 meters and a transfer rate of 2.1Mbps, whereas Wi-Fi covers a range of 150 meters with a transfer rate of 31.4Mbps (nominal net throughput of 802.11g with CSMA/CA enabled). Green circles around the nodes in Fig. 7 represent coverage ranges.

All nodes reserve 5MB of their memory for the purposes of message caching, with the exception of trams. For the first two experiments we configured tram nodes to use 50MB of memory to cache messages, while we repeated the third experiment multiple times varying the cache size. The values we used for the cache size are 5MB, 10MB, 15MB, 25MB, and 50MB.

In our simulations we considered five different applications and, for simplicity, we assumed that each application produces and consumes messages only in the context of a single dedicated ADC, i.e., there is a 1:1 mapping between five different applications and five different ADCs. Each node simultaneously runs a variable number of randomly chosen applications, ranging from one to five, which is set according to a negative exponential random distribution. To maintain constant the number and type of applications running on each node, we fed a pseudorandom number generator with the same seed values across all simulations. Thus, the tuple [68, 25, 15, 10, 2] describes the number of nodes per number of applications running on each node in all our experiments. This means that 68 nodes run one application chosen randomly among the five available applications, 25 nodes run two randomly chosen applications, and so on. The same nodes will join the same set of ADCs across all simulations. Note that the sum of all elements in the tuple is 120, because the 6 trams do not take part either in message generation or in its consumption. Said otherwise, in our simulations, trams are not eligible sources or destinations for any message, they do not join any ADC, but they might still contribute to message routing, forwarding, and caching according to the ADC mode configured and the dissemination and caching strategies employed.

7.2. First experiment: Parameter Determination

We measured performance in our simulations by collecting statistics and computing the network delivery ratio, delivery delay, overhead, the number of dropped messages, and the number of discarded messages. We classify messages as dropped if they were first stored in one node's caching memory and then deleted in a future occasion, in accordance with some cache eviction policy, whereas we talk about discarded messages in case they were rejected at the moment of reception because of the chosen ADC mode.

We define the network delivery ratio d_N as follows:

$$d_N = \frac{D_N}{\sum_n \sum_m S_{n,m}}, \quad (1)$$

where D_N is the number of messages correctly delivered to their destination, and the term at the denominator is the sum over all nodes of all messages they subscribed to during the time of the simulation (excluding any warm-up time). In compliance with this definition, the value of the single term $S_{i,j}$ is 1 if and only if node i has subscribed to the content carried within message j , or 0 otherwise. Defined this way, the relation $0 \leq d_N \leq 1$ holds.

Similarly, we define the network overhead o_N as follows:

$$o_N = \frac{T_N - D_N}{D_N + 1}, \quad (2)$$

where T_N is the total number of transmissions performed by all nodes in the network, including collisions, and D_N is the same as above. Thus defined, we have $0 \leq o_N$.

The first experiment aims at identifying the optimal values of FP and CP for the SP-ADC mode in the simulated scenario. Their optimal values depend on several parameters, including the specific network configuration, the number of nodes, their mobility, and their cache size. Determining a network-level optimal pair for FP and CP is very important for at least two reasons. First, it allows us to fix those variables to well-determined values and thus compare SP-ADC against S-ADC and U-ADC. Equally importantly, the process of discovering the best values of FP and CP gives us the chance to investigate their impact on the communications performance and on each different network statistic.

In order to identify the optimal parameter values, we set the dissemination strategy to SnWBR and the ADC mode to SP-ADC and we used several combinations for the FP (accepting one value among 0.1, 0.3, 0.5, 0.7, and 0.9) and CP (whose value could be one among 0.1, 0.3, 0.5, 0.7, 0.9, and 1.0) parameters. Incidentally, we observe that a value of 0 for both FP and CP corresponds to S-ADC, while a value of 1.0 for both the parameters corresponds to U-ADC; we will compare SP-ADC to those modes thoroughly in our second experiment. The first experiment required a total of 30 simulation runs to be carried out completely. After we completed data collection, we analyzed the network delivery ratio and network overhead metrics achieved in each simulation.

Fig. 8 shows the trend of delivery ratio (a) and overhead (b) for all simulation runs. Interestingly, the simulation that scored the highest delivery ratio (0.8055) was configured with FP equal to 0.1 and CP equal to 0.5, while the one with both FP and CP set to 0.1 showed the lowest overhead, as one might expect. However, what is most important is to find pairs of values that allow reaching delivery ratios close to the optimum and, at the same time, maintaining the overhead under control. For this purpose, we designed the following score function:

$$Sf_N = \frac{d_N^2}{\sqrt[3]{(1 + o_N)}}, \quad (3)$$

where d_N and o_N are defined as in (1) and (2), respectively. The unit is added to the network overhead value so that the radicand is always higher than or equal to 1. We chose to use a power of two at the numerator and a root of degree 3 at the denominator to give a higher importance to differences in the network delivery ratio than in the network overhead. In Fig. 9 we plotted the value of (3) for all pairs of FP and CP obtained in the simulations run in our first experiment.

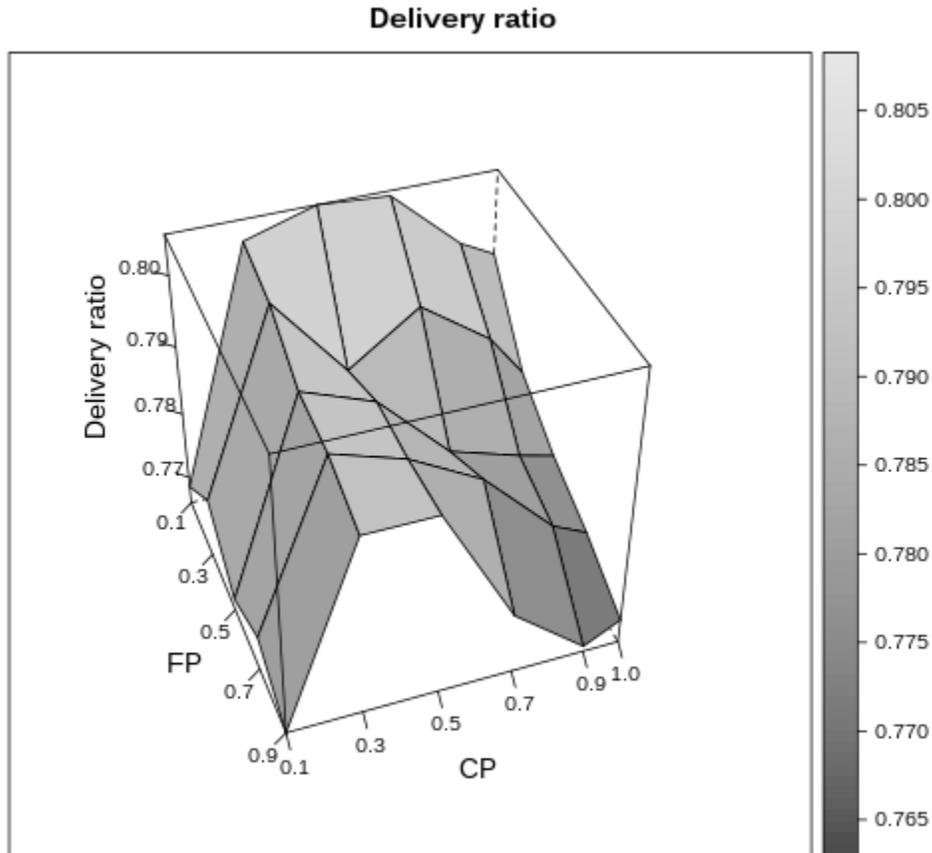


Fig. 8a. Trend of delivery ratio varying the FP and CP parameters in our first experiment.

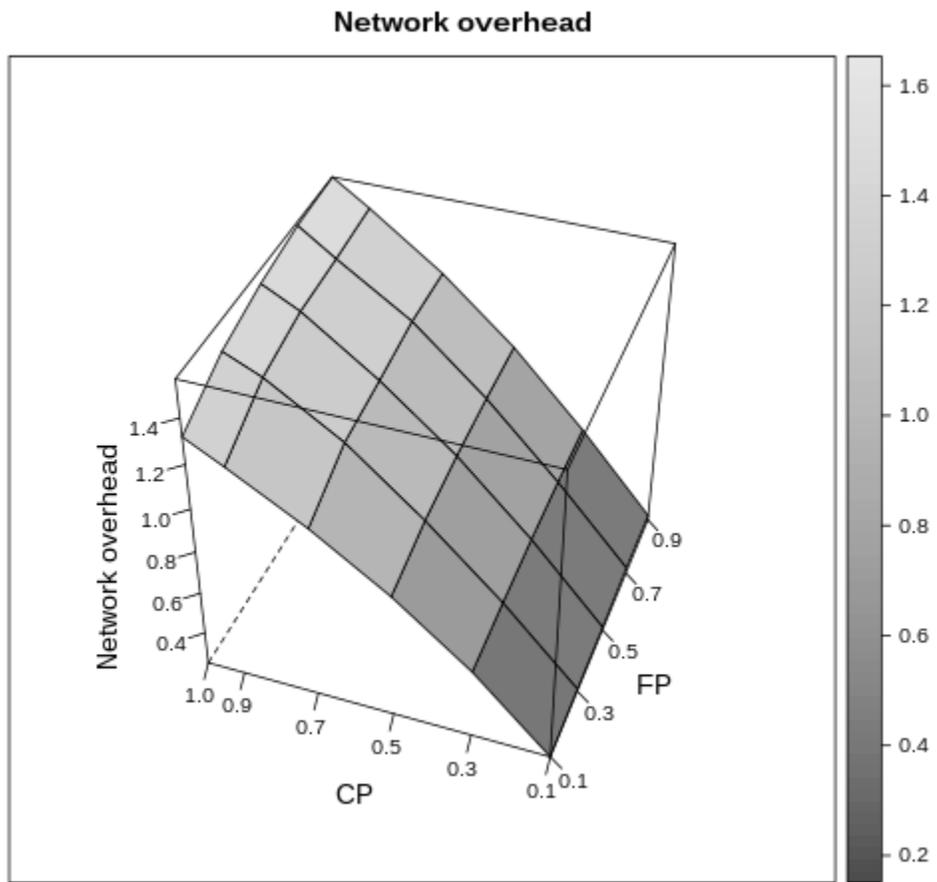


Fig. 8b. Trend of the measured network overhead varying the values of the FP and CP parameters in our first experiment.

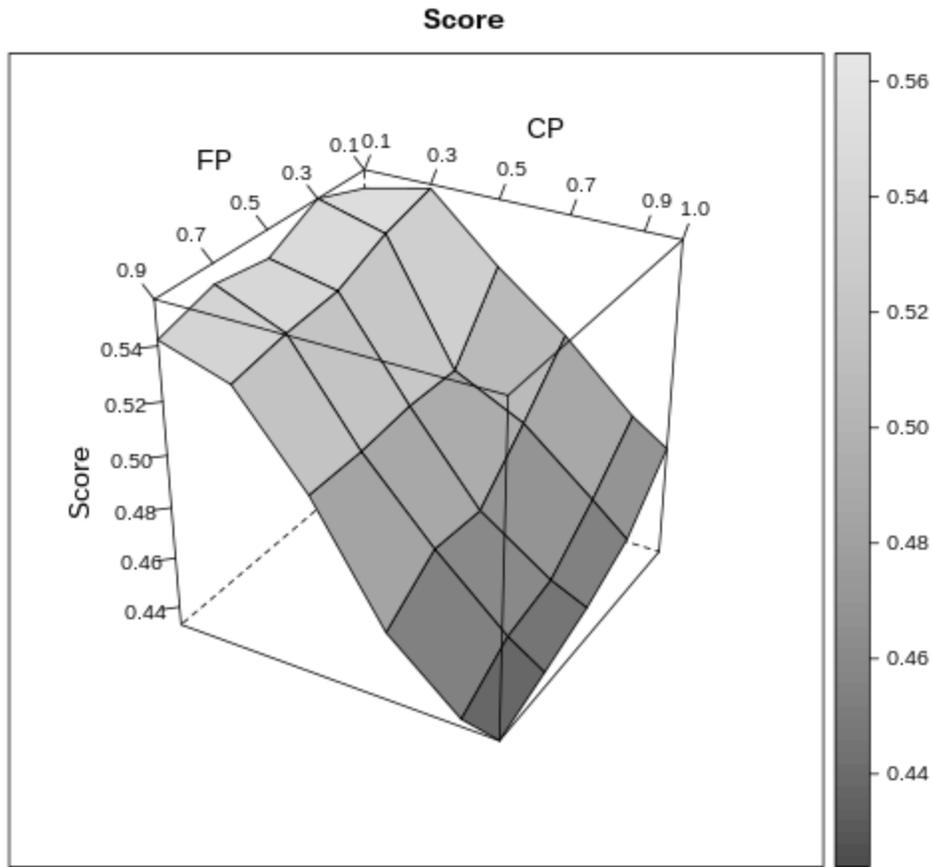


Fig. 9. The values of the score function with the changing of the FP and CP parameters.

The simulation configured with FP set to 0.3 and CP set to 0.1 is associated with the highest score, $Sf_N = 0.5547$, and produced a high delivery ratio (0.8026) while keeping the overhead down to 0.5661. In our next experiments we will fix FP and CP to these values. The measured metrics also show the strong impact of FP and CP on the performance of communications and first exhibit the potential of SP-ADCs. Analyzing the other statistics, we notice that the average network latency stays comparable across all simulations. The lowest peak measured was 1173 seconds, which corresponds to values of 0.7963 and 1.2317 for d_N and o_N , respectively, and that was obtained setting FP to 0.1 and CP to 0.9. On the other hand, the highest latency found was 1313 seconds, achieved by setting FP to 0.9 and CP to 1.0, which corresponds to a network delivery ratio of 0.7693 and a network overhead of 1.5610.

7.3. Second experiment: Comparison of ADC modes

In our second experiment, we ran 6 simulations using the scenario illustrated in Section 7.1 and all possible combinations of dissemination strategy (EBR and SnWBR) and ADC mode (Strict, Semipermeable, and Unconstrained). As stated above, we configured the two simulations that used the Semipermeable mode with the values 0.3 and 0.1 for FP and CP, respectively. The

reader might note that the results obtained with EBR/S-ADC and with SnWBR/S-ADC correspond: this is the direct consequence of restraining message dissemination only among their destinations, against which both routing algorithms behave the same way.

Table 1 compares the network delivery ratio (as defined by eq. (1) above, in percentage), the median of the delivery delay (in seconds), the network overhead (as defined by eq. (2) above), and the total number of transmissions, message drops, and discarded messages for all 6 simulations. As one might expect from the characteristics of the two dissemination strategies, the values of delivery ratio and average latency obtained using the EBR are slightly better than those obtained using the SnWBR, whereas the SnWBR achieved the lowest overhead. However, the most interesting aspect is analyzing their trend when varying the ADC mode. In fact, SP-ADC reaches the highest delivery ratio, which is counterintuitively much better than that measured using the unconstrained strategy. At the same time, SP-ADC keeps the overhead ratio under control and it does not affect the average latency significantly.

We believe that the explanation lies in the more frugal use of the available resources: reducing the number of total transmissions enabled a much better exploitation of the scarce bandwidth and decreased collisions significantly: results show -68.0 and -59.2 percent of collisions against a total number of message transmissions of -46.1 and -36.8 percent using the EBR and the SnWBR, respectively, when comparing Semipermeable and unconstrained ADC strategies. In addition, SP-ADC reduced the number of message drops (-42.9 percent for EBR and -34.2 percent for SnWBR) compared to their unconstrained variants, hence a more efficient usage of the cache. At the same time, moving from S-ADC to SP-ADC increased the delivery ratio of 9.56 and 9.21 percent and reduced the expected delivery delay by 21.6 and 20.4 percent, using the EBR and the SnWBR respectively.

Table 1. Summary of the results of all simulations performed for the second experiment.

	Delivery Ratio	Delivery Delay (s)	Network Overhead	Num of Transmissions	Num of Collisions	Num of Dropped Messages	Num of Discarded Messages
EBR + S-ADC	71.05%	975.2	0.07	47335	5927	43681	29215
EBR + SP-ADC	80.61%	764.7	0.73	86557	15915	78447	34159
EBR + U-ADC	77.1%	628.9	2.35	160653	49657	137377	0
SnWBR + S-ADC	71.05%	975.2	0.07	47335	5927	43681	29215
SnWBR	80.26%	776.6	0.57	78227	16701	72842	27581

+ SP-ADC							
SnWBR + U-ADC	76.86%	839.4	1.57	123028	40790	110251	0

7.4. Third experiment: Evaluation of Trams' Contribution

So far, we treated tram nodes exactly like any other node with respect to the ADCs. The goal of our third and last experiment is to better appreciate their potential for the purposes of message dissemination. In fact, given their high mobility, greater network and storage resources, and access to an external, virtually infinite, power source, tram nodes represent a strategic resource to improve performance in next-generation communications scenarios [93]. To achieve our goal, we decided to collect statistics running multiple simulations where tram nodes have a special “*” subscription, which specifies their interest in messages from any channel. This way, trams can take a more active part in message routing, regardless of the routing algorithm and the ADC mode chosen. In addition, with this last experiment, we tried to achieve a more profound understanding of the impact that the size of the cache on tram nodes has on the overall system performance. Therefore, we fixed the routing algorithm to SnWBR and we ran 15 simulations to test all possible combinations of the 3 ADC modes and 5 different values for tram nodes' cache size: 5MB, 10MB, 15MB, 25MB, and 50MB (as explained in section 7.1). Again, we configured FP and CP in the SP-ADC to 0.3 and 0.1, respectively.

Fig. 10a compares the trends of the delivery ratio obtained by each ADC mode with varying cache sizes on the trams. As one would easily expect, greater memory resources correspond to higher delivery ratios regardless of the ADC mode used. The Semipermeable mode again achieved the best result, with a ratio of 0.8446, while using the Strict and Unconstrained modes yielded ratios of 0.8154 and 0.8026, respectively. Note that 0.8446 is the highest delivery ratio measured across all simulations we performed, with an increase of 4.78 percent with respect to the value obtained using SP-ADC on top of the EBR in our second experiment, which was the highest score reached so far.

On the other hand, Fig. 10b shows that the cache size does not seem to have a strong impact on the overhead ratio, which largely depends on the ADC mode used. The highest values we measured are 0.0894 for the S-ADC, 0.4576 for the SP-ADC, and 1.3183 for the U-ADC. However, the comparison of these results with those measured in our previous experiment is very interesting. In fact, the increase in the delivery ratio shown above is associated to a reduction in the average network overhead: the SP-ADC and the U-ADC scored -19.72 and -16.03 percent, respectively. On the contrary, the S-ADC showed a growth of 27.71 percent in the measured network overhead. This suggests that exploiting tram nodes can also reduce bandwidth consumption effectively, when either the SP-ADC or the U-ADC is used.

Fig. 10c plots the average message delivery latency measured while varying the cache size equipped on the tram nodes. All curves are U-shaped; those obtained using the S-ADC and the

SP-ADC have their minimum with a cache size of 15MB, while the U-ADC yielded the lowest latency with a cache size of 10MB. For small values of cache size, the U-ADC mode seems to obtain better results, but the latency grows very quickly with the increasing of cache size. On the other hand, the SP-ADC shows higher latency values with small caches, but it grows more slowly than the Unconstrained mode, thereby leading to lower latencies with larger memories.

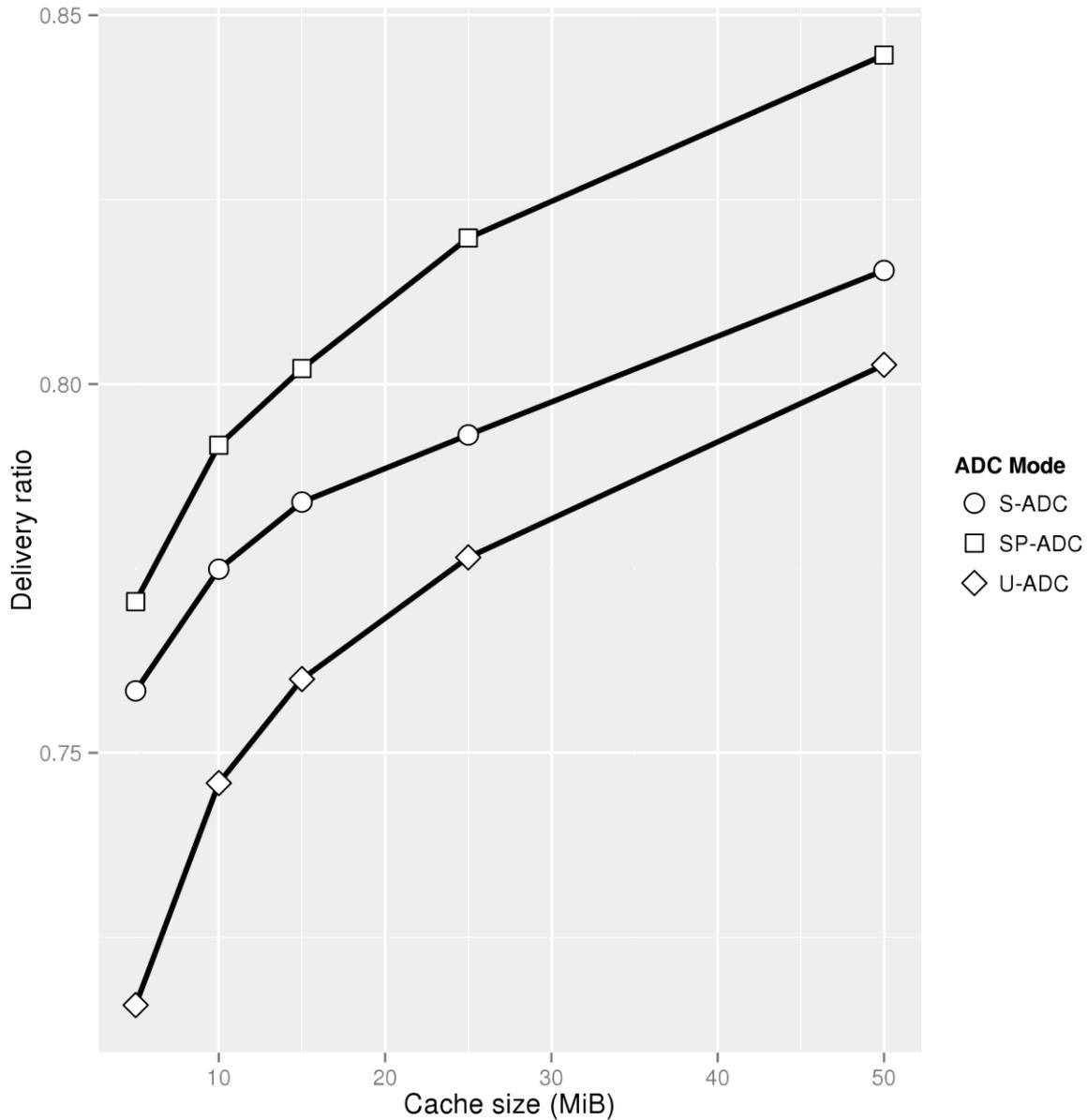


Fig. 10a. Delivery ratio scores for each Application-level Dissemination Channel mode, varying the cache size on tram nodes.

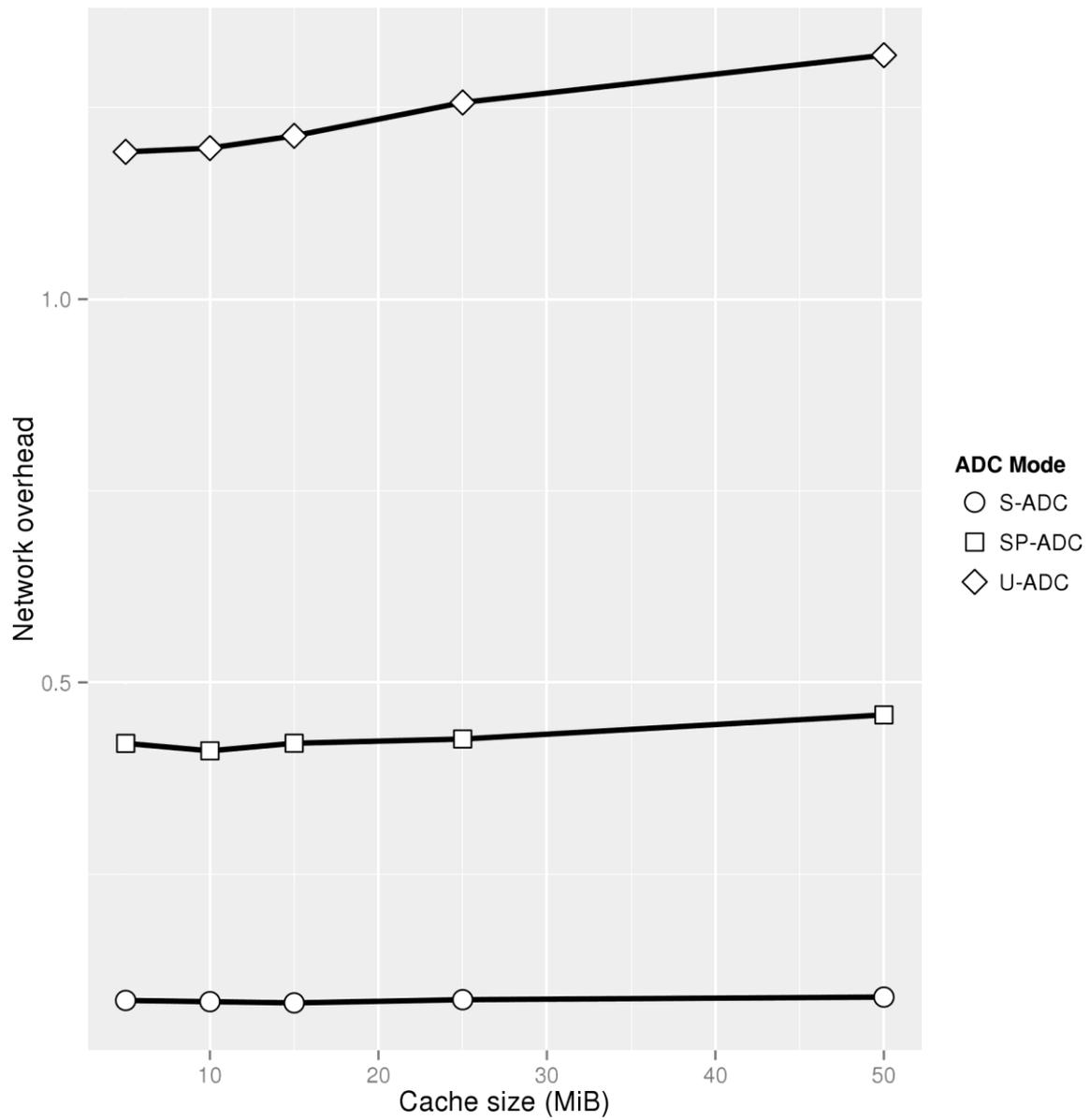


Fig. 10b. Network overhead for each Application-level Dissemination Channel mode, varying the cache size on tram nodes.

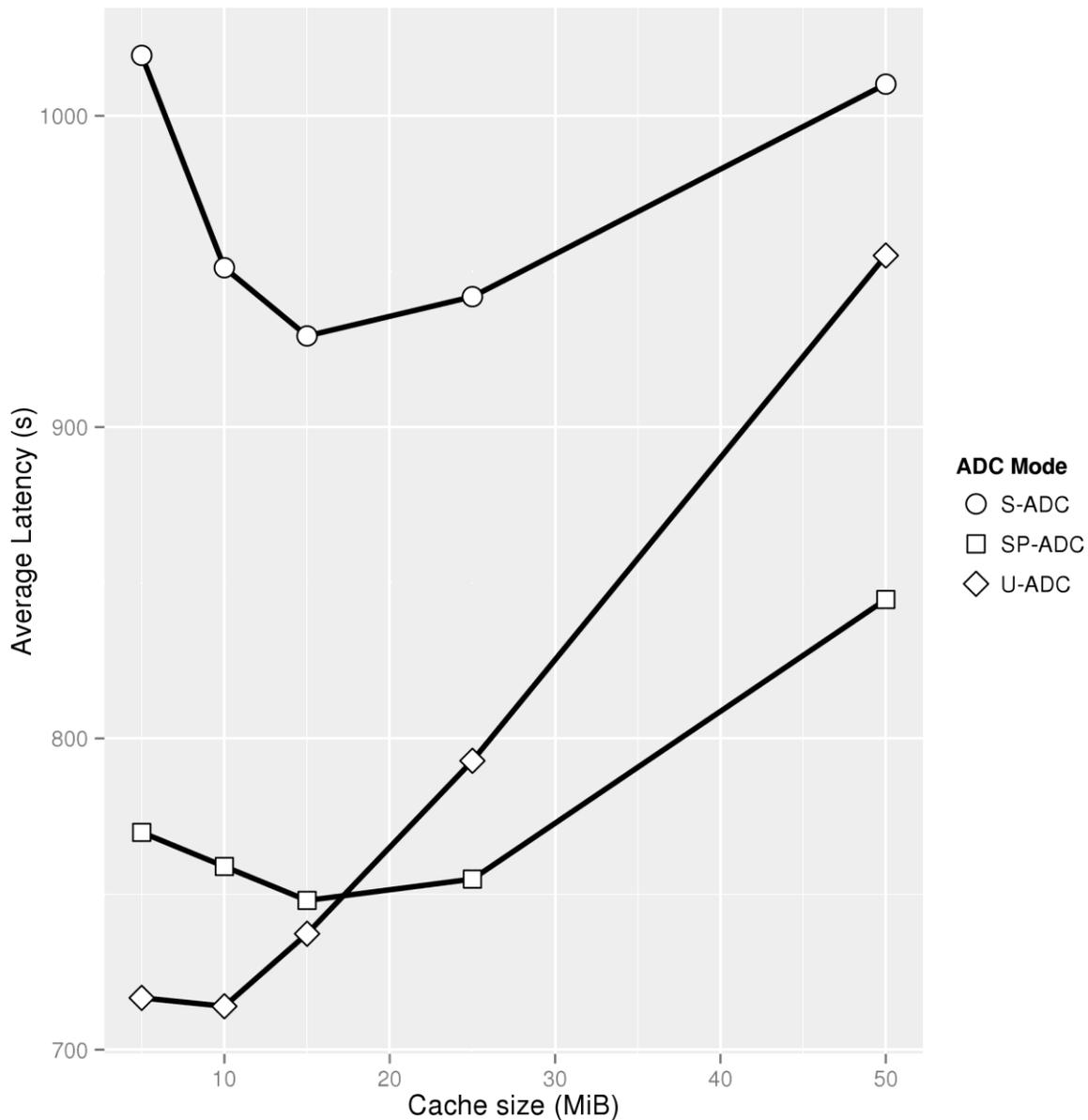


Fig. 10c. Average message delivery latency for each Application-level Dissemination Channel mode, varying the cache size on tram nodes.

8. Conclusions and Future Work

Our work on ICeDiM has shown the potential of the ICN paradigm in the next-generation communications scenario. In particular, devising and developing an ICN communications middleware that provides the right set of tools and methodologies for the management of resources acquires a special relevance in this context, where the amount of bandwidth, memory, and computational power available in the network changes continuously and their

inefficient use could seriously undermine the correct and efficient functioning of the complete system. Despite the promising results obtained by ICeDiM in a simulated smart city scenario, the techniques presented in this paper are not meant to be final or exhaustive, and their refinement requires further research.

In the future, we are interested in extending ICeDiM to collect a history of encountered nodes and to compute statistics about the distribution of IOs across the network. The information generated this way could then be shared with nearby nodes, e.g., by appending it to HELLO messages or by generating and broadcasting a new type of message. We expect that this new, acquired knowledge will enable nodes to build up an internal representation of the state of the network and, based on it, improve the decision-making process relative to IO caching and dissemination. For instance, caching decisions concerning an IO could be made based on its availability across the network, or the decision to evict an IO from the cache could be influenced by the probability of encountering the destination node before the expiration of the TTL, computed from the history of node contacts.

Another interesting research direction concerns the design of an algorithm for the automated tuning of the FP and CP parameters in ICeDiM, so that the permeability of Semipermeable ADCs can be adapted to the current number, type, and duration of the communication opportunities. We believe that such an approach would increase the effectiveness of SP-ADCs in some critical cases, like when networks become very sparse, e.g., at the edge of smart cities, or very dense, e.g., in case of social events, important sport competitions, concerts, flash mobs, and so on.

Finally, there is the obvious need to port and/or integrate existing applications to the networking environments of the next-generation. This suggests the deployment of solutions, such as node- or network-level proxies, that can bridge the gap between applications and the communication middleware while enabling applications to achieve good levels of performance and reliability [26] [94].

References

[1] G. Benincasa, A. Morelli, C. Stefanelli, N. Suri, M. Tortonesi, "Agile Communication Middleware for Next-generation Mobile Heterogeneous Networks", IEEE Software, Vol. 31, No. 2 (Special Issue on Next Generation Mobile Computing), pp. 54-61, March-April 2014.

[2] N. Suri, G. Benincasa, M. Tortonesi, C. Stefanelli, J. Kovach, R. Winkler, R. Kohler, J. Hanna, L. Pochet, S. Watson, "Peer-to-Peer Communications for Tactical Environments: Observations, Requirements, and Experiences", IEEE Communications Magazine, Vol. 48, No. 10 (Special Issue on Military Communications), pp. 60-69, October 2010.

[3] G. Carofiglio, G. Morabito, L. Muscariello, I. Solis, M. Varvello. "From content delivery today

to information centric networking”, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 57, No. 16, pp. 3116-3127, November 2013.

[4] Yongqiang Huang, Hector Garcia-Molina, “Publish/subscribe in a mobile environment”, *Wireless Networks*, Vol. 10, No. 6, pp. 643-652, November 2004.

[5] M. Amadeo, C. Campolo, A. Molinaro, G. Ruggeri, “Content-centric wireless networking: A survey”, *Computer Networks*, Vol. 72, pp. 1-13, 29 October 2014.

[6] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, “A survey of information-centric networking”, *IEEE Communications Magazine*, Vol. 50, No. 7, pp. 26-36, July 2012.

[7] D. Perino, M. Varvello, “A Reality Check for Content Centric Networking”, *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking (ICN '11)*, pp. 44-49, 2011

[8] G. Zhang, Y. Li, T. Lin, “Caching in information centric networking: A survey”, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 57, No. 16, pp. 3128-3141, November 2013.

[9] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, I. Stoica, “A Data-oriented (and Beyond) Network Architecture”, *ACM SIGCOMM Computer Communication Review*, Vol. 37, No. 4, pp. 181-192, October 2007.

[10] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard, “Networking Named Content”, *International Conference on Emerging Networking Experiments and Technologies (CoNEXT '09)*, pp. 1-12, 2009.

[11] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. C. Claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, “Named Data Networking”, *ACM SIGCOMM Computer Communication Review*, Vol. 44, No. 3, pp. 66-73, July 2014.

[12] B. Ahlgren, M. D'Ambrosio, M. Marchisio, I. Marsh, C. Dannewitz, B. Ohlman, K. Pentikousis, O. Strandberg, R. Rembarz, V. Vercellone, “Design Considerations for a Network of Information”, *ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT '08)*, No. 66, pp. 1-6, 2008.

[13] N. Fotiou, D. Trossen, G. C. Polyzos, “Illustrating a Publish-Subscribe Internet Architecture”, *Journal of Telecommunications Systems*, Vol. 51, No. 4, pp. 233-245, December 2012.

[14] S. Singh, H.S. Dhillon, J.G. Andrews, "Offloading in Heterogeneous Networks: Modeling, Analysis, and Design Insights", *IEEE Transactions on Wireless Communications*, Vol. 12, No. 5, pp. 2484-2497, May 2013

[15] K. Wei, G. Mao, W. Zhang, Y. Yang, Z. Lin, C. S. Chen, "Optimal Microcell Deployment for Effective Mobile Device Energy Saving in Heterogeneous Networks", in Proceedings of IEEE International Conference on Communications (ICC 2014), Sydney, Australia, Jun 2014.

[16] A. Asadi, Q. Wang, V. Mancuso, "A Survey on Device-to-Device Communication in Cellular Networks," IEEE Communications Surveys & Tutorials, Vol. 16, No. 4, pp. 1801-1819, 24 April 2014.

[17] Y. Sambo, M. Shakir, F. Héliot, M. Imran, S. Mumtaz, K. Qaraqe, "Device-to-Device Communication in Heterogeneous Networks", S. Mumtaz, J. Rodriguez (Eds.) "Smart Device to Smart Device Communication", Springer, 2014.

[18] N. Mitton, S. Papavassiliou, A. Puliafito, K. S. Trivedi, "Combining Cloud and sensors in a smart city environment", EURASIP Journal on Wireless Communications and Networking, August 2012, 2012:247.

[19] Hans Schaffers, Nicos Komninos, Marc Pallot, Brigitte Trousse, Michael Nilsson, Alvaro Oliveira, "Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation", The Future Internet, Lecture Notes in Computer Science Volume 6656, 2011, pp. 431-446.

[20] I. Stojmenovic, S. Wen, "The Fog computing paradigm: Scenarios and security issues", in Proceedings of 2014 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1-8, 2014.

[21] T. Luan, L. Gao, Z. Li, Y. Xiang, L. Sun, "Fog Computing: Focusing on Mobile Users at the Edge", arXiv preprint, Technical Report accessible at: <http://arxiv.org/pdf/1502.01815.pdf>.

[22] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014–2019", White Paper, pp. 1-42. February 3, 2015, available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf.

[23] J. Lee, Y. Yi, S. Chong, Y. Jin, "Economics of WiFi Offloading: Trading Delay for Cellular Capacity", IEEE Transactions on Wireless Communications, Vol. 13, No. 3, pp. 1540-1554, March 2014

[24] N. Suri, E. Benvegna, M. Tortonesi, C. Stefanelli, J. Kovach, J. Hanna, "Communications Middleware for Tactical Environments: Observations, Experiences, and Lessons Learned," IEEE Communications Magazine, Vol. 47, No. 10, pp. 56-63, October 2009.

[25] K.-L. Chiu, Y.-S. Chen, R.-H. Hwang, "Seamless session mobility scheme in heterogeneous wireless networks", International Journal of Communication Systems, Vol. 24, No. 6, pp. 789-809, June 2011.

[26] M. Tortonesi, A. Morelli, C. Stefanelli, R. Kohler, N. Suri, S. Watson, "Enabling the deployment of COTS applications in tactical edge networks," *IEEE Communications Magazine*, Vol. 51, No. 10, pp. 66-73, October 2013.

[27] Z. Zhuang, T.-Y. Chang, R. Sivakumar, and A. Velayutham, "Application-Aware Acceleration for Wireless Data Networks: Design Elements and Prototype Implementation", *IEEE Transactions on Mobile Computing*, Vol. 8, No. 9, September 2009.

[28] S. Eum, Y. Shoji, M. Murata, N. Nishinaga, "Design and implementation of ICN-enabled IEEE 802.11 access points as nano data centers", *Journal of Network and Computer Applications*, Vol. 50, No. C, pp. 159-167, April 2015.

[29] P. Mendes, "Combining data naming and context awareness for pervasive networks", *Journal of Network and Computer Applications*, Vol. 50, pp. 114-125, April 2015.

[30] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, M. Handley, "How hard can it be? Designing and implementing a deployable multipath TCP", in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation (NSDI'12)*, pp. 29-42, 2012.

[31] M. Louta, P. Bellavista, "Bringing Always Best Connectivity Vision a Step Closer: Challenges and Perspectives", *IEEE Communications Magazine*, Vol. 51, No. 2, pp. 158-166, February 2013.

[32] G. Tyson, N. Sastry, I. Rimal, R. Cuevas, A. Mauthe, "A survey of mobility in information-centric networks: challenges and research directions", *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications (NoM '12)*, pp. 1-6, 2012.

[33] W. K. Chai, D. He, I. Psaras, G. Pavlou, "Cache 'Less for More' in Information-Centric Networks (Extended Version)", *Computer Communications*, Vol. 36, No. 7, pp. 758-770, 1 April 2013.

[34] G. Tyson, N. Sastry, R. Cuevas, I. Rimal, A. Mauthe, "Where is in a Name? A Survey of Mobility in Information-Centric Networks", *Communications of the ACM (CACM)*, Vol. 56, No. 12, 2013.

[35] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos, "A Survey of Information-Centric Networking Research", *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 2, pp. 1024-1049, 19 July 2013.

[36] A. Vasilakos, Z. Li, G. Simon, W. You, "Information centric network: Research challenges and opportunities", *Journal of Network and Computer Applications*, Vol. 52, pp. 1-10, June

2015.

[37] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, J. Wilcox., "Information-centric networking: seeing the forest for the trees", Proceedings of the 10th ACM Workshop on Hot Topics in Networks (HotNets-X), Article 1, New York, NY, USA, 2011.

[38] S. Kapadia, B. Krishnamachari, L. Zhang, "Data Delivery in Delay Tolerant Networks: A Survey", Mobile Ad-Hoc Networks: Protocol Design, Prof. Xin Wang (Ed.), 30 January 2011

[39] T. Spyropoulos, T. Turetli, K. Obraczka, "Routing in Delay Tolerant Networks Comprising Heterogeneous Node Populations", IEEE Transactions on Mobile Computing, Vol.8, No. 8, pp. 1132-1147, August 2009.

[40] A. Mtibaa, K. Harras, "CAF: Community aware framework for large scale mobile opportunistic networks", Computer Communications, Vol. 36, No. 2, pp. 180-190, 2013.

[41] D. Karamshuk, C. Boldrini, M. Conti, A. Passarella, "Human mobility models for opportunistic networks", IEEE Communications Magazine, Vol. 49, No. 12, pp. 157-165, December 2011.

[42] C. Boldrini, M. Conti, A. Passarella, "Exploiting users social relations to forward data in opportunistic networks: the HiBOp solution", Pervasive and Mobile Computing, 2008.

[43] C. Boldrini, M. Conti, A. Passarella, "Design and performance evaluation of ContentPlace, a social-aware data dissemination system for opportunistic networks", Computer Networks, Vol. 54, No. 4, pp. 589-604, March 2010.

[44] A. Martín-Campillo, J. Crowcroft, E. Yoneki, R. Martí, "Evaluating opportunistic networks in disaster scenarios", Journal of Network and Computer Applications, <http://dx.doi.org/10.1016/j.jnca.2012.11.001>, 2012.

[45] L. Pelusi, A. Passarella, M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks", IEEE Communications Magazine, Vol. 44, No. 11, pp. 134-141, November 2006.

[46] M. Conti, S. Giordano, M. May, A. Passarella, "From opportunistic networks to opportunistic computing", IEEE Communications Magazine, Vol. 48, No. 9, pp. 126-139, September 2010.

[47] C.M. Huang, K.C. Lan, C.Z. Tsai, "A Survey of Opportunistic Networks", Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops (AINAW '08), pp. 1672-1677, 2008.

[48] H. A. Nguyen, S. Giordano, "Routing in Opportunistic Networks", International Journal of Ambient Computing and Intelligence (IJACI), Vol. 1, No. 3, 2009.

- [49] A. Vahdat, D. Becker, "Epidemic routing for partially connected ad hoc networks", Technical Report CS-200006, Duke University, 2000.
- [50] T. Spyropoulos, K. Psounis, C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks" Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WDTN '05), pp. 252-259, 2005.
- [51] A. Lindgren, A. Doria, O. Schelén, "Probabilistic routing in intermittently connected networks", ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 7, No. 3, pp. 19-20, July 2003.
- [52] J. Burgess, B. Gallagher, D. Jensen, B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking", Proceedings of the IEEE INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 2005.
- [53] T. Spyropoulos, K. Psounis, C. S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility", Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW '07), pp. 79-85, 19-23 March 2007.
- [54] C. Boldrini, M. Conti, J. Jacopini, A. Passarella, "HiBOp: a History Based Routing Protocol for Opportunistic Networks", Proceedings of WOWMOM, pp.1-12, 2007.
- [55] H. A. Nguyen, S. Giordano, "Spatiotemporal routing algorithm in opportunistic networks", 2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), pp. 1-6, 2008.
- [56] C. Boldrini, M. Conti, A. Passarella, "Impact of social mobility on routing protocols for Opportunistic Networks", IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007), pp. 1-6, 2007.
- [57] M. Amadeo, C. Campolo, A. Molinaro, "Forwarding strategies in named data wireless ad hoc networks: Design and evaluation", Journal of Network and Computer Applications, Vol. 50, pp. 148-158, April 2015.
- [58] A. Barzan, B. Bonne, P. Quax, W. Lamotte, M. Versichele, N. V. d. Weghe, "A comparative simulation of opportunistic routing protocols using realistic mobility data obtained from mass events", IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), pp. 1-6, 2013.
- [59] M. Varvello, I. Rimac, U. Lee, L. Greenwald, V. Hilt, "On the Design of Content-Centric MANETs," in Proceedings of the 8th international conference on Wireless On-Demand Network Systems and Services (WONS), January 26-28, 2011.

- [60] S. Baseer, M.I Channa, K. Ahmed, "A Review of Routing Protocols of Heterogeneous Networks", International Journal of Computer Applications (IJCA), Vol. 2, No. 2, pp. 58-66, 2010.
- [61] Y. Kim, I. Yeom, "Performance analysis of in-network caching for content-centric networking", Computer Networks, Vol. 57, No. 13, pp. 2465-2482, 9 September 2013
- [62] Y. Li, H. Xie, Y. Wen, Z. Zhang, "Coordinating In-Network Caching in Content-Centric Networks: Model and Analysis," Proceedings of the 33rd International Conference on Distributed Computing Systems (ICDCS '13), pp. 62-72, 2013.
- [63] Y. Wang, Z. Li, G. Tyson, S. Uhlig, G. Xie, "Optimal Cache Allocation for Content-Centric Networking", in Proceedings of IEEE International Conference on Network Protocols, 2013.
- [64] A. Dabirmoghaddam, M. Mirzazad-Barijough, J. J. Garcia-Luna-Aceves, "Understanding Optimal Caching and Opportunistic Caching at "The Edge" of Information-Centric Networks", in Proceedings of ICN 2014.
- [65] D. Kim, S.-W. Lee, Y.-B. Ko, J.-H. Kim, "Cache capacity-aware content centric networking under flash crowds", Journal of Network and Computer Applications, Vol. 50, pp. 101-113, April 2015.
- [66] K. Suksomboon, S. Tarnoi, Y. Ji, M. Koibuchi, K. Fukuda, S. Abe, M. Nakamura, M. Aoki, S. Urushidani, and S. Yamada, "PopCache: Cache more or less based on content popularity for information-centric networking", Proceedings of 38th IEEE Conference on Local Computer Networks (LCN 2013), pp. 236-243, 2013.
- [67] I. Psaras, W. K. Chai, G. Pavlou, "Probabilistic In-Network Caching for Information-Centric Networks", 2nd ACM SIGCOMM Workshop on Information-Centric Networking (ICN 2012), pp. 55-60, Helsinki, Finland, August 2012.
- [68] M. Draxler, H. Karl, "Efficiency of On-Path and Off-Path Caching Strategies in Information Centric Networks", IEEE International Conference on Green Computing and Communications (GreenCom) 2012, pp. 581-587, 20-23 November 2012
- [69] M. Dehghan, A. Seetharam, T. He, T. Salonidis, J. Kurose, D. Towsley, "Optimal Caching and Routing in Hybrid Networks", Proceedings of IEEE Military Communications Conference (MILCOM), Baltimore, MD, USA, October 2014.
- [70] E. Monticelli et al., "Combining Opportunistic and Information Centric Networks in Real World Applications", NetSys 2015.
- [71] M. Amadeo, A. Molinaro, "CHANET: A content-centric architecture for IEEE 802.11

MANETs", 2011 International Conference on the Network of the Future (NOF), pp.122-127, Paris, 28-30 Nov. 2011.

[72] Q. Ayub, S. Rashid, Zahid, A. H. Abdullah, "The optimization of Spray and Wait routing Protocol by prioritizing the message forwarding order", International Journal of Innovation and Applied Studies, Vol. 3, No. 3, pp. 794–801, July 2013.

[73] N. Suri, M. Tortonesi, G. Benincasa, R. Lenzi, C. Stefanelli, L. Sadler, "Exploring Value of Information-based Approaches to Support Effective Communications in Tactical Networks", in IEEE Communications Magazine, Vol. 53, No. 10 (Special Feature on Military Communications), pp. 39-45, October 2015.

[74] H. Geng, N. Jamali, "Supporting many-to-many communication", Proceedings of the 2013 workshop on Programming based on actors, agents, and decentralized control (AGERE! 2013), pp. 81-86, 2013.

[75] D. Hughes, G. Coulson, J. Walkerdine, "A Survey of Peer-to-Peer Architectures for Service Oriented Computing", In Handbook of Research on P2P and Grid Systems for Service-Oriented Computing: Models, Methodologies and Applications, ed. N. Antonopoulos, G. Exarchakos, M. Li, A. Liotta, pp. 1-19, 2010.

[76] A. Bęben, J. Mongay Batalla, W. K. Chai, J. Śliwiński, "Multi-criteria Decision Algorithms for Efficient Content Delivery in Content Networks", Annals of Telecommunications, Special Issue on Networked Digital Media, Vol. 68, No. 3-4, pp. 153-165, April 2013.

[77] K. Katsaros, C. Wei, W. Ning, G. Pavlou, H. Bontius, M. Paolone, "Information-centric networking for machine-to-machine data delivery: a case study in smart grid applications," IEEE Network, Vol. 28, No. 3, pp. 58-64, May-June 2014.

[78] S. Y. Oh, D. Lau, M. Gerla, "Content Centric Networking in Tactical and Emergency MANETs," IFIP Wireless Days (WD), pp. 1-5, Venice, Italy, 2010.

[79] A. Detti, D. Tassetto, N. Blefari-Melazzi, F. Fedi, "Exploiting Content Centric Networking to Develop Topic-based, Publish-Subscribe MANET Systems", Elsevier Ad Hoc Networks, Vol. 24, Part B, pp. 115-133, January 2015.

[80] S. Wood, J. Mathewson, J. Joy, M.-O. Stehr, K. Minyoung, A. Gehani, M. Gerla, H. Sadjadpour, J.J. Garcia-Luna-Aceves, "ICEMAN: A System for Efficient, Robust and Secure Situational Awareness at the Network Edge", in IEEE Military Communications Conference (MILCOM), pp. 1512-1517, 2013.

[81] E. Yoneki, P. Hui, S. Chan, J. Crowcroft, "A socio-aware overlay for publish/subscribe communication in delay tolerant networks", Proceedings of the 10th ACM Symposium on Modeling, analysis, and Simulation of Wireless and Mobile systems (MSWiM '07), 2007.

- [82] N. Bin, N. Santhapuri, Z. Zifei, S. Nelakuditi, "Routing with opportunistically coded exchanges in wireless mesh networks," 2nd IEEE Workshop on Wireless Mesh Networks - WiMesh 2006, pp. 157-159, 25-28 Sept. 2006.
- [83] A. Boukerche, A. Darehshoorzadeh, "Opportunistic Routing in Wireless Networks: Models, Algorithms, and Classifications", ACM Computer Surveys, Vol. 47, No. 2, Article 22, pp. 1-36, November 2014.
- [84] A. Detti, B. Ricci, N. Blefari-Melazzi, "Mobile peer-to-peer video streaming over information-centric networks", Computer Networks, Vol. 81, pp. 272-288, 22 April 2015.
- [85] L. Militano, M. Condoluci, G. Araniti, A. Molinaro, A. Iera, F.H.P. Fitzek, "Wi-Fi Cooperation or D2D-based Multicast Content Distribution in LTE-A: a Comparative Analysis", IEEE International Conference on Communications Workshops (ICC), pp.296-301, 10-14 June 2014.
- [86] L. Hogie, P. Bouvry, F. Guinand, "An Overview of MANETs Simulation", Journal of Electronic Notes in Theoretical Computer Science (ENTCS), Vol. 150, No. 1, pp. 81-101, March 2006.
- [87] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, H. Zedan, "A Comprehensive Survey on vehicular Ad Hoc network", Journal of Network and Computer Applications, Vol. 37, pp. 380-392, January 2014
- [88] M. Liu, Y. Yang, Z. Qin, "A Survey of Routing Protocols and Simulations in Delay-Tolerant Networks", Proceedings of the 6th International Conference on Wireless Algorithms, Systems, and Applications (WASA '11), pp. 243-253, 2011.
- [89] A. Keränen, J. Ott, T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation", in Proceedings of 2nd International Conference on Simulation Tools and Techniques (SIMUTools'09), Rome, No. 55, pp. 1-10, March 2009.
- [90] IEEE Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, available online at: <http://standards.ieee.org/getieee802/download/802.11-2012.pdf>.
- [91] A. Huang, C. Lea, A. K. Wong, "A joint solution for the hidden and exposed terminal problems in CSMA/CA wireless networks", Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 56, No. 14, pp. 3261-3273, September 2012.
- [92] J.C.-P. Wang, M. Abolhasan, D. R. Franklin, F. Safaei, "Characterising the Behaviour of IEEE 802.11 Broadcast Transmissions in Ad Hoc Wireless LANs", ICC '09 - IEEE International

Conference on Communications, pp.1-5, 14-18 June 2009.

[93] A. Morelli, C. Stefanelli, M. Tortonesi, N. Suri, "Mobility Pattern Prediction to Support Opportunistic Networking in Smart Cities," 2013 International Conference on MOBILE Wireless MiddleWARE, Operating Systems and Applications (Mobilware), pp.166-175, 11-13 Nov. 2013.

[94] A. Morelli, R. Lenzi, C. Stefanelli, N. Suri, M. Tortonesi, "A Proxy Gateway Solution to Provide QoS in Tactical Networks and Disaster Recovery Scenarios", 11th ACM International Symposium on QoS and Security for Wireless and Mobile Networks (ACM Q2SWinet 2015) – ACM MSWiM 2015, Cancún, Mexico, 2-6 November 2015.

[95] A. Socievole, E. Yoneki, F. De Rango, J. Crowcroft, "ML-SOR: Message routing using multi-layer social networks in opportunistic communications", Computer Networks, Vol. 81, No. 22, pp. 201-219, April 2015.

[96] R.-I. Ciobanu, R.-C. Marin, C. Dobre, V. Cristea, "Interest-awareness in data dissemination for opportunistic networks," Ad Hoc Networks Journal, Vol. 25, pp. 330-345, February 2015.

[97] W. A. Moreira Junior, P. Mendes, S. Sargento, "Social-aware Opportunistic Routing Protocol based on User's Interactions and Interests", International Conference on Ad Hoc Networks, pp. 100-115, July 2014.

[98] C. Velásquez-Villada, Y. Donoso, "Delay/Disruption Tolerant Networks based message forwarding algorithm for rural Internet connectivity applications", 6th International Conference on Computers Communications and Control (ICCCC), pp. 16-22, May 2016.