Journal of
*Sensor and
Actuator Networks*

MDPI

Article

Routing Protocols for Delay Tolerant Networks: A Reference Architecture and a Thorough Quantitative Evaluation

Khalil Massri [1], Andrea Vitaletti [2,*], Alessandro Vernata [2] and Ioannis Chatzigiannakis [2]

[1] College of Information Technology, Computer Science Departement, Hebron University, Hebron, 00970 West Bank, Palestine; khalil.massri@gmail.com

[2] Dipartimento Di Ingegneria Informatica Automatica e Gestionale (DIAG), Università degli Studi di Roma "La Sapienza", 00185 Rome, Italy; alex.vernata@gmail.com (A.V.); ichatz@dis.uniroma1.it (I.C.)

* Correspondence: vitaletti@dis.uniroma1.it; Tel.: +39-06-7727-4026

Academic Editor: Dharma P. Agrawal
Received: 12 January 2016; Accepted: 21 March 2016; Published: 29 March 2016

Abstract: In this paper, we propose a reference architecture for Delay-Tolerant Networking (DTN) routing protocols and a thorough quantitative evaluation of many protocols proposed in the literature. We categorize DTN protocols according to their use of the three techniques that are the key elements of our reference architecture: *queue management*, *forwarding*and *replication*. Queue management orders and manages the messages in the node's buffer; forwarding selects the messages to be delivered when there is a contact; and finally, replication bounds the number of replicas in the network. Contrary to most previous papers, where either only qualitative comparisons have been presented or only a single category of protocols has been analyzed, in our work, we discuss the results of our experimental activity on many of the DTN protocols in the literature. Our results, which have been obtained both using synthetic and real mobility traces, show that an effective combination of the proposed techniques can significantly improve the performance of the protocols in terms of delivery ratio, overhead and delay.

Keywords: delay-tolerant networks; routing protocols; performance evaluation; experimentation

1. Introduction

In the Internet of Things era, every one of over a trillion everyday items will include at least some ability to store and process information; additionally, and more importantly, sharing that information over the global Internet with the other trillion items. The technological goal is to integrate the Internet and the web with everyday objects (such as doors, chairs, electric appliances, cars, *etc.*) and ultimately interconnect the digital and physical domains. Clearly, the types of objects to be connected with the Internet, e.g., in terms of usage, size and numbers, is extremely diverse, thus having different computation and communication requirements. For this reason, a large number of computing architectures and networking paradigms have been proposed, and different networking standards have been developed.

In most cases, the operational and performance characteristics of the newly-introduced technologies make conventional (Internet-like) networking approaches either unworkable or impractical. Concepts of occasionally-connected networks have become a very common approach for a very diverse range of real-world applications; real-world cases that suffer from frequent partitions and that rely on more than one divergent set of protocols or protocol families. The dominant approach of networking in such cases is to provide the nodes with significant memory capabilities, so that messages can be stored for long periods of time. Whenever communication is established, the stored messages are opportunistically forwarded to the connecting node.

Almost a decade ago, Fall *et al.* [1] proposed an architecture, using the term "Delay-Tolerant Networks (DTN)", that utilized asynchronous message forwarding to achieves interoperability between various types of DTNs. Since then, the Internet Research Task Force (IRTF) established the DTN Research Group (DTNRG) [2], in an attempt to standardize the architecture in RFC4848 [3] and to introduce a communication protocol called the Bundle Protocol (described in RFC5050 [4]) that lies on top of the transport layer, to support interoperability between heterogeneous networks, intermittent connectivity, long or variable delay for DTNs.

Since then many different protocols have been proposed in order to establish efficient communication among the nodes of the networks [5]. The DTN architecture has been used in a variety of real-world pilot applications, including mobile sensor networks for pervasive information gathering [6], human-centric sensing networks, where nodes are carried by humans [7,8], networks comprised of ferry boats, where nodes are positioned in sea vessels moving along predefined routes over large geographic regions [9], and inter-planetary networks, designed to withstand the noise and delays incurred by communication across astronomical distances, to account for the occlusion of satellites and planned communication windows [10]. In [11], a number of applications of DTN protocols in urban pervasive use cases is presented.

1.1. The Need for a Reference Architecture

Unfortunately, after all of the research efforts committed to DTNs during the past few years, only an extremely limited number of DTN protocols became part of mainstream ICT systems. A main reason for this gap between protocol design and wide uptake is the diversity in contexts in which DTN routing is applicable, thus making different protocols suitable only for specific cases and hard to understand with respect to how they can be applied to different application scenarios.

In the past few years, researchers have attempted to survey and organize the literature by providing a unifying taxonomy of DTN routing protocols. However, the approach of existing surveys for classifying DTN routing protocols is to focus on specific aspects of the opportunistic operation of the network. This approach provides taxonomies that are very helpful for better understanding the literature. Unfortunately, they provide an isolated list of features, and in doing, so they do not help the application development process by providing insights on how to combine these techniques for better results.

In this paper, we try to overcome this problem by following a different approach and by placing special emphasis on the design aspects of DTN routing protocols. We identify the three most fundamental components existing in DTN routing protocols. We consider *forwarding*, *replication* and *queue management* as the main techniques for DTN protocols, and we show how each protocol can be classified according to the techniques it adopts. Indeed, these techniques have been already considered in the literature, yet our approach is unifying and, in our opinion, better clarifies their roles and interaction. In this sense, we propose a *reference architecture for DTN routing protocols*.

DTN routing protocols can vary widely, based on the techniques employed to store messages, the approaches introduced for selecting the next node to forward, *etc.* Interestingly, our reference architecture is broad enough to describe almost all solutions examined. Based on our reference architecture, we classify a selected set of very relevant DTN routing protocols. We believe that the proposed architecture allows a clear identification of the main techniques characterizing DTN routing protocols.

Based on the given reference architecture, we show that each DTN protocol proposed in the literature is either based on a *single technique* or on the *combination of techniques*. As a consequence, each protocol can be classified in our taxonomy according to the techniques it adopts. In particular, we started considering the techniques proposed in the taxonomy presented in [12], and we identified *forwarding* and *replication* as the most distinguished ones. Furthermore, in light of the work of Lenas *et al.* [13], we also consider *queue management* as a fundamental technique.

Our focus is on the combination of strategies for storing, forwarding and replicating messages and how they affect the overall performance of basic communication. Providing secure communication in a DTN protocol is certainly a challenging problem, since the operation of the network is based on the fundamental aspect of trust among the nodes. Guaranteeing that messages will be delivered even in the case where malicious nodes participate in the exchange process or protecting the network against the injection of fake messages are certainly difficult (if not impossible) tasks. Providing end-to-end encrypted communication can be achieved given that the sender and receiver nodes have exchanged a common set of cryptographic keys directly without involving other nodes. In our architecture, the source/destination authentication mechanisms, protection against tampering of message contents and detection of replay can be incorporated in the storing, forwarding and replication mechanisms. These aspects are not within the scope of this paper to investigate, and the interested reader may look into [14,15].

1.2. The Need for a Quantitative Evaluation

A second reason for the limited uptake of DTN protocols in industrial ICT systems is the approach followed by the majority of the DTN research community in terms of evaluating the performance of the protocol designs proposed. As real-world deployments of DTN or the creation of large-scale test-beds are costly and hard to control, the majority of the proposed protocols are evaluated using simulated environments. It is well known that simulated environments lack the necessary realism and provide results that could lead to great surprises when deployed in real-world environments. As an attempt to improve the accuracy of simulation studies, one part of the research community increased the complexity of the simulated environments in order to make them more realistic, while in parallel, another part focused on the development of complex synthetic mobility models [16], as well as the collection of mobility and connectivity traces from real-world networks [17,18].

All efforts so far have led to a wide spectrum of scenarios and tools, spanning from completely synthetic scenarios (like random way-point) to very realistic (based on trace data), from generic network simulators to very complex and specific to a particular networking technology. Since the research community did not converge to a specific range of benchmarking scenarios and simulation tools, it is very hard (if not impossible) to make a comparative evaluation of different DTN designs based on a literature review. Evaluations performed in very realistic settings are very specific to that particular scenario, and any conclusions drawn cannot be expected to be valid in other scenarios. This is also evident by examining the existing surveys that only provide a qualitative comparison of the protocols; as an example, see Table 2 in [12] and Tables 1–3 in [19]. It is therefore important to provide an evaluation framework that will produce comparable quantitative evidence.

In this paper, we attempt to provide a detailed quantitative comparison framework that establishes a repeatable set of benchmarking scenarios. The scenarios proposed are based on realistic technical assumptions that are derived from existing real-world applications and address both indoor and outdoor deployments. Our evaluation is based both on the synthetic scenario, as well as traces collected from real-world deployments with varying mobility patterns and network area size. We work with the Opportunistic Network Environment (ONE) simulator [20], a JAVA-based simulator for evaluating protocols for opportunistic networking. We selected the ONE simulator as it is most commonly used in the relevant literature and for which existing implementations are available for most of the protocols considered. Although a recently-developed tool, it has gained wide acceptance in the DTN research community and is considered as the reference simulator for DTN [21]. We significantly extend ONE by implementing two new synthetic mobility models (Levy Walk [16] and Community [6,22]) and a mechanism for reproducing the motion of nodes based on real traces of mobility (Infocom2006 [23], Rome Taxis [24]). Furthermore, we implement for the first time three DTN routing protocols (Fuzzy Spray [25], Fad [6] and Scar [26]). Our model and protocol implementations are open-source and available online for free. We believe that our implementation effort reinforces ONE and works towards making it a mature tool for the DTN research community [27].

Given this quantitative comparison framework, we conduct a detailed evaluation of 10 characteristic DTN routing protocols proposed in the relevant literature. Our results are based on extensive simulations with the ONE simulator [20], which is considered one of the reference simulator for

DTN [21]. The selection of the 10 protocols is based on the techniques employed: we attempted to selected as many different techniques as possible in order to establish a broader understanding of the implications of each technique on the the delivery ratio, the delay of message delivery and the overhead imposed. We strongly believe that studying such a broad range of techniques and protocols allows one to get a well-rounded understanding of the challenges, weaknesses and benefits of existing DTN routing protocols, while at the same time highlighting different approaches that can be explored by future research. We note that in the past, several attempts have been made to quantitatively evaluate the performance of DTN protocols; however, very few studies incorporated a large and diverse set of protocols, with the most notable one being [28].

The rest of the article is structured as follows. In the following section, we present the high-level architecture and basic functionality of the main components of our reference design. The section also includes the implementation aspects of 16 main-stream DTN protocols in relation to the proposed reference architecture. In Section 3, we present the quantitative experimental evaluation conducted on 10 representative DTN protocols. The section provides a detailed description of the benchmarking scenario, both in terms of synthetic traces and traces acquired from real-world experiments. An extensive discussion of the results acquired is included along with a comparative study of the performance of the protocols. In Section 4, we provide technical recommendations in terms of the mechanisms introduced in the literature and how they can be applied in practical scenarios. Finally, in Section 5, we summarize our findings and provide pointers for future research.

2. Reference Architecture

In this section, we present a reference architecture for DTN routing protocols. The basic components of the architecture are presented in the following section. Then, we examine a selected number of characteristic DTN routing protocols and discuss how they can be incorporated

2.1. Basic Components

The internal architecture of a DTN node is shown in Figure 1. Upon contact, a node exchanges a summary of the information necessary to update its knowledge (*i.e.*, a local view) on the environment in which it operates. As an example, it can exchange the list of messages in the queue or the list of prior encountered nodes. This knowledge drives all of the subsequent actions and is used by queue management (QM) to assign a rank to the messages in the queue. If the queue is full, queue management applies a suitable eviction policy to make room for new incoming messages. The forwarding policy (FW) selects the messages to be forwarded, which are possibly duplicated (Replication (R) in order to improve the effectiveness of the protocol.

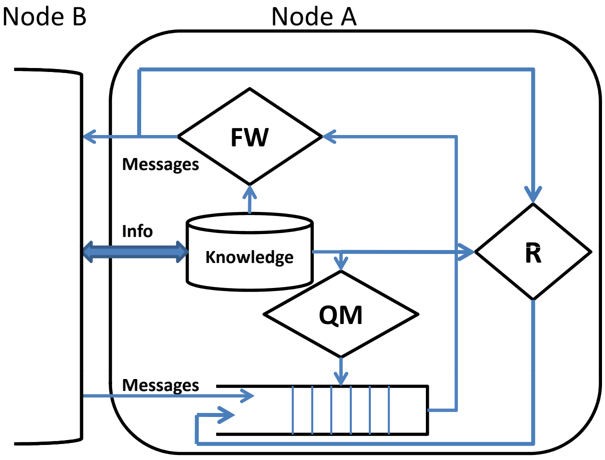


Figure 1. Node architecture.

Queue Management (QM) defines a total ordering on the messages in the queue on the basis of the node's knowledge. QM orders all of the messages, also those ones that will not be taken into consideration as possible candidates to be forwarded. When new messages must be accommodated in the queue and there is no room, the eviction policy selects the messages to be discarded according to that ordering. The intrinsic delay of DTN applications implies that messages can be enqueued for a long time. Due to the limited buffer size of DTN nodes, an efficient queuing management policy must be adopted to avoid the eviction of important messages. In some cases, more than one queuing policy is combined to produce more efficient message ordering [29]. In the following, we briefly discuss the main QM policies:

- First-In-First-Out (FIFO) is the default QM policy.
- Destination independent, where QM uses only parameters that are not related to the destination, such as message hop count [29], forward transmission count and message size [25], and other types listed in [30].
- Destination dependent: In this case, QM uses parameters dependent on the destination. In [29], the cost to reach the destination is computed using Dijkstra, while [18] estimates the delay. The work in [6] uses the message fault tolerance, which is an indicator of the importance of the fact that a message reaches its destination (this is related to the redundancy of the message).

Forwarding (FW) selects the subset of messages in the queue to be forwarded. In most cases, the messages are selected on the basis of the same ordering defined by the QM, but in some cases [18,29], a new ordering among messages can be employed. Contrary to what happens for QM, this new ordering is defined only with respect to the current contact and lasts only for the contact time (*i.e.*, the time the two nodes can communicate). In other words, the new ordering is only used to define a priority on the messages to be forwarded to the current contact. Briefly, the main FW policies are:

- Direct-delivery: The message is only delivered to the destination (*i.e.*, no intermediate nodes are employed).
- Always: Upon contact, the message is always forwarded. This policy requires low computational power, but can lead to a very high number of transmissions.
- Knowledge based: Messages are forwarded based on the current knowledge in terms of contextual, historical or social information. Contextual information regards the knowledge that can be inferred by the node only based on its current status, such as battery level, speed, direction, *etc.* Historical information is obtained over time and is used to evaluate future behaviors of the network; inter-contact time, history of encounters, age of encounters and contact time are examples of such information. Finally, considering that in many DTN scenarios, nodes are carried by humans, social information is used to describe the relationships among the users in order to predict social behaviors that can be used to improve the effectiveness of the forwarding policy.

Replication (R) controls and bounds the number of copies of a message in the network and is used to increase the robustness of the protocols. Observe that a message that has been selected by the FW policy to be delivered to the current contact is normally deleted from the queue, but the R policy can enqueue it again in order to have multiple copies in the network.

- Single copy: Messages are never replicated. Once a message has been delivered to an encounter, it is deleted from the queue.
- Limited: The total number of copies of a message in the network is bounded.
- Controlled: A message is replicated only if a condition holds.
- Unlimited: There are no constraints on the number of replicas in the network.

2.2. Implementation Aspects of DTN Routing Protocols

We now discuss existing DTN routing protocols presented in the literature and how their internal components are mapped to the proposed reference architecture. We carefully examine their proposed techniques and classify them accordingly. The result of this effort is shown in Table 1. We remark that the list of protocols examined here is by no means exhaustive. Apart from the protocols discussed below, there exist other protocols, such as, for example, Spray-and-Focus [31], GeoSpray [32], Encounter-based Routing [33], GeOpps [34], *etc.*, that have been studied extensively by the relevant research community and are also compatible with our reference architecture. Our selection is based on the mechanisms introduced by the protocols and to demonstrate how diverse mechanisms can be accommodated by our reference architecture.

Direct-Delivery and First-Contact [20]: Both are single copy routing protocols, namely only one copy of each message exists in the network. In Direct-Delivery, messages are only delivered to their final destination. First-Contact always forwards the messages to the first contact and then deletes them from the queue; this means that messages can be handled by several nodes. Both protocols employ FIFO queuing.

Epidemic [35]: Messages are broadcast to all neighbors. When there is no room in the message queue, the oldest messages are evicted. Messages are always forwarded according to a FIFO policy, and no bound on the number of replicas is considered (*i.e.*, R is unlimited).

Spray-and-Wait [36]: This protocol is made of two phases. The spray phase (only once): L message copies are initially spread to L distinct "relays". The wait phase: if the destination is not reached in the spray phase, the L nodes carrying a message copy perform direct transmission. The forwarding technique is a mix of always (Spray) and direct delivery (Wait). The replication is limited (the bound is L), and the queue management is FIFO. *PRoPHET [37]:* This one uses delivery predictability as a metric to evaluate the likelihood of one node reaching the destination. The delivery predictability is updated when a new node is encountered or the time out expires. A node carrying a message delivers it to all of the neighbors with a delivery predictability higher than its own. The queuing and replication techniques are the same as Epidemic (FIFO and unlimited); but forwarding is based on delivery predictability, and it is thus history based. Similar considerations on forwarding hold for protocols like FRESH [38], which uses last encounter age, and SEPR [39], which evaluates the shortest path.

Fuzzy-Spray [25]: The protocol is based on two parameters: the Forward Transmission Count (FTC) and the message size. These parameters are input to a fuzzy rule, which prioritizes the messages to be transmitted. Selected messages are broadcast to all of the neighbors. The forwarding and replication techniques are the same as Epidemic (always and unlimited). The queue management is destination independent and is based on the message priority.

SCAR [26]: Context and history information, such as co-location, mobility and battery level, are used to calculate the delivery predictability of each neighbor, and messages are delivered to the neighbors with the highest delivery predictability first. For each message, there is a single master and L backup copies generated by the source node. Messages are ordered in the queue according to their master/backup label, and master messages are forwarded first. Backup copies can be evicted, while master ones are never evicted. In this case, forwarding is a mix of context and history based; replication is limited; and queue management is destination independent.

FAD [6]: Similar to PRoPHET, FAD forwards the messages to the neighbors with the highest delivery probability. However, in this case, messages are ordered in the queue according to their fault tolerance. Fault tolerance is proportional to the number of replicas in the network and also takes into consideration the delivery probability. Messages with low fault tolerance are forwarded first in order to increase the number of copies in the network of messages with a low probability of reaching the destination. In this case, forwarding is history based, while queue management is destination dependent, and replication is unlimited.

MaxProp [29]: Each node has a routing table, which predicts the likelihood (*i.e.*, the cost) to reach another node in the future through its current neighbors. Routing tables are updated on the basis of the information obtained by the neighbors, and messages are ordered and forwarded according to their cost to reach the destination. As such, forwarding is history based, and replication is unlimited. The queuing technique is a combination of destination dependent (Dijkstra to evaluate the cost to reach the destination) and destination independent (the current number of hops that a message has accumulated).

RAPID [18]: Most DTN protocols aim at maximizing the probability of finding a path between the source and the destination, neglecting all of the other relevant metrics, such as delay, energy consumption, *etc.* On the contrary, RAPID tries to optimize a specific routing metric (e.g., worst-case delivery delay). RAPID treats DTN routing as a resource allocation problem. It translates routing metrics into per-message utilities, which determine how messages should be replicated in the system. When there is a need to make room for new messages, messages with lower utility are deleted first. The forwarding technique is history based; messages with higher marginal utility are forwarded first, and replication is unlimited. Queue management is destination dependent; messages are prioritized according to their utility value (*i.e.*, the estimated delay to reach the destination).

Cluster-Based Routing [22]: The basic idea of this protocol is to distributively group mobile nodes with similar delivery probability (*i.e.*, nodes that are frequently in contact) into a cluster. Nodes in a cluster can interchangeably share their resources (such as buffer space) for overhead reduction and load balancing. This protocol uses single copy replication and FIFO queuing, and its forwarding technique is history based (cluster based).

NETCAR [40]: The scheduler of NETCAR uses the neighborhood index (contact history) to select the messages to be forwarded. Message replication is controlled using parameters that come from the context (buffer occupancy, maxTTL, minEpidemicLevel, maxEpidemicLevel, *etc.*). Message aging and the number of replicas are used to evict messages. NETCAR uses a history-based forwarding with controlled replication and destination independent queue management.

ORWAR [41]: This protocol uses context information, such as speed, direction, radio range and bandwidth, to estimate the contact window size. Only messages that can be transmitted in such an interval are forwarded. A message utility function (defined by the application), the so-called "utility per bit ratio", ranks the messages in the queue and also regulates the number of replicas for each message. Forwarding is context based; replication can be considered controlled, and queue management is destination independent.

HiBOp [42]: In HiBOp, a mix of context and history information is used to forward messages only to nodes with a higher delivery probability. This knowledge is also used to control the amount of message replicas (a measure of redundancy and fault tolerance for the message is computed). Only the source can replicate a message; the other nodes can only forward it. The number of replicas is calculated such that the probability of losing all of the replicas is below the threshold specified by the application. This implies that replication is controlled, and forwarding is a mix of context and history based.

BubbleRap [7] and SimBet [8]: These are both social-based protocols. BubbleRap combines the knowledge on the community structure and the centrality of the nodes to decide whether to forward. Each message has two types of ranking, global and local (related to its community). Forwarding is done using the global ranking, until the message reaches a node that is in the same community of the destination node. Then, local ranking is used until the destination is reached or the message expires. Each node forwarding a message does not delete its copy unless the message is delivered to the community of its destination. In this case, forwarding is social based, and replication is unlimited. In SimBet, similarity and betweenness utility functions are combined to select the message to be forwarded, and there is only a single copy of the message in the network (*i.e.*, no replication).

Table 1. Protocol classification. QM, queue management; FIFO, First-In-First-Out.

		Protocols [1]															
		Direct Contact	**First Contact**	**E**	**PRoPHET**	**Spray-and Wait**	**Fuzzy Spray**	**SCAR**	**FAD**	**MaxProp**	**RAPID**	**Cluster Based**	**NETCAR**	**ORWAR**	**HiBOp**	**SimBet**	**BubbleRap**
FW	Direct Del.	√	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-
	Always	-	√	√	-	√	√	-	-	-	-	-	-	-	-	-	-
	Context	-	-	-	-	-	-	√	-	-	-	-	-	√	√	-	-
	History	-	-	-	√	-	-	√	√	√	√	√	√	-	√	-	-
	Social	-	-	-	-	-	-	-	-	-	-	-	-	-	- √	√	
R	Single	√	√	-	-	-	-	-	-	-	-	√	-	-	-	√	-
	Limited	-	-	-	-	√	-	√	-	-	-	-	-	-	-	-	-
	Controlled	-	-	-	-	-	-	-	-	-	-	-	√	√	√	-	√
	Unlimited	-	-	√	√	-	√	-	√	√	√	-	-	-	-	-	-
QM	FIFO	√	√	√	√	√	-	-	-	-	-	√	-	-	√	√	√
	Destination Depend.	-	-	-	-	-	-	-	√	√	√	-	-	-	-	-	-
	Destination Independ.	-	-	-	-	-	√	√	-	√	-	-	√	√	-	-	-

[1] Direct-Delivery Protocol [20]; First-Contact Protocol [20]; Epidemic Protocol [35]; PRoPHET Protocol [37]; Spray-and-Wait Protocol [36]; Fuzzy-Spray Protocol [25]; SCAR Protocol [26]; FAD Protocol [6]; MaxProp Protocol [29]; RAPID Protocol [18]; Cluster-Based Routing Protocol [22]; NETCAR Protocol [40]; ORWAR Protocol [41]; HiBOp Protocol [42]; SimBet Protocol [8]; BubbleRap Protocol [7].

3. Performance Evaluation

We now proceed by conducting an extensive comparative study of 10 recently-presented, characteristic DTN routing papers (see Section 2.2). The performance evaluation is based on simulated experiments based on a reference benchmarking scenario. We use (1) synthetic mobility traces (based on three well-established models) and (2) real-world traces to evaluate the performance of the protocols in environments that represent both (i) indoor and (ii) outdoor cases. The combination of synthetic/real-world mobility traces, indoor/outdoor cases and the operating parameters of the high-level user application (for generating traffic) constitute a thorough benchmarking scenario. The benchmarking scenario and routing protocols are implemented in The Opportunistic Network Environment simulator (ONE) [20]. To the best of our knowledge, this is the first attempt to quantitatively compare the performance of such a large selection of DTN protocols.

We here note that the ONE simulator already includes implementations for Direct-Delivery, First-Contact, Epidemic, Spray-and-Wait, PRoPHET and MaxProp. We implement the other four protocols, Fuzzy-Spray, FAD, SCAR and RAPID, by extending the ONE simulator. In order to validate the truthfulness of the protocol implementations, we conduct a set of small-scale experiments based on well-defined mobility and message generation patterns. In the sequel, we reproduced the evaluation scenario included in the original publications and validated that our protocol implementations provide similar results to the ones described by the original authors. In order to facilitate this process, we used real-time visualizations of the network execution for simulations of small-sized networks in combination with debugging messages, describing in detail the simulated execution of each protocol. We are confident that through this process, we fine-tuned the operation of all protocols, and our implementations achieve high truthfulness compared to the original protocols.

3.1. Reference Benchmarking Scenario

We examine the performance of each protocol and classify its performance using a reference scenario that is general enough to address specifications derived from real-world applications. We consider that nodes that participate in the opportunistic network are smart phones carried by humans, and as such, they do not have control on their mobility (*i.e.*, passively mobile). In terms of wireless communication technology for *ad hoc* connectivity, we consider 802.15.1 (Bluetooth), which is supported by the reference implementation for DTN, as proposed in [2]. According to Bluetooth specifications, transmission speed is about 0.2 Mbps = 25.00 kBps and transmission range about 10 m.

To evaluate the effects of the buffer size on the performance, we combined different message and buffer sizes. In the small buffer size case, there are an average of 70 messages in the queue, while in the larger case, the messages are about 700. We vary the number of nodes from 10–70 nodes moving in the reference area. Each node generates a new message to a random destination every 1–3 min. Messages never expire, namely their TTL is set to infinity. These technical specification have been derived from the application scenario presented in [1,12,19,43,44].

We consider two different cases for the area covered by the network nodes (*i.e.*, within which humans are moving):

Indoor environment: a rectangular area of 400 m × 400 m that may represent a conference hall or an office space;

Outdoor environment: a rectangular area of 8 km × 8 km that represents a city center or a town. The area is further analyzed using a 32 × 32 grid, with each grid cell covering a square area of 250 m ×250 m [24].

Clearly, the way in which nodes move within this area is a critical factor that affects the performance of the network [45,46]. Essentially, movement affects nodes' opportunity to meet other nodes, that is the probability to reach the destination and the inter-contact probability. Consequently, the messages' likelihood to be delivered before being dropped depends on the pattern of the mobility of the nodes.

In order to evaluate this effect, we incorporate two approaches: (1) synthetic mobility patterns: we use three mobility models for simulating mobility by humans, a group of humans and robots; (2) real-world traces of mobility: we use traces acquired from real-world experiments in order to reproduce the mobility of the nodes. In all cases considered, the average speed of nodes (used in different mobility models and as observed in the real-world traces) for the indoor environment is chosen to be in the range of walking speed (2.8 . . . 7.2 km/h), while for the outdoor environment, to be in the range of driving speed in the city center (5 . . . 50 km/h), with an average speed of 12.29 km/h [24].

More specifically, the mobility patterns considered are the following:

- *Random Waypoint (RWP)* [20]: In RWP, nodes randomly choose the next destination and move there with a random speed. Upon arrival at the destination, the node pauses for a while and then chooses a new destination. This mobility pattern is suitable for simulating the mobility of robots.
- *Levy Walk (LW)* [16]: This mobility model tries to emulate human walk patterns in outdoor environments. The next direction is uniformly selected at random, while distance from the current position and speed are selected according to a power law distribution. In this model, the authors extended their preliminary work in [47], where they proposed a mobility model called SLAW (Self-similar Least Action Walk) in order to produce synthetic walk traces containing features of human walk traces.
- *Community (CM)* [6,22]: This model emulates communication within groups of people. The network area is divided into the so-called communities, and each node has a probability to get in/out of a community, as well as transit from one community to another [22]. Our implementation follows the specification given in [6].
- *Real-world traces, indoor environment*: We use the Haggle traces [23], a dataset on a human mobility experiment lasting four days at Infocom 2006. A total of 78 volunteers joined the experiment carrying an iMote device with a Bluetooth radio (about 30 m of range) used to log contacts between users.
- *Real-world traces, outdoor environment*: We use the Rome Taxi traces [48], a dataset on a set of 320 taxi cabs monitored for 30 days, from 1 February–2 March of 2014. Each taxi driver has a tablet equipped with a GPS device that periodically (7 s) retrieves the GPS position and sends it to a central server.

The settings used to run our simulations are summarized in Table 2.

Table 2. Simulation settings.

Parameter	Indoor Environment	Outdoor Environment
Simulation time	43,200 s (12 h)	30 days
Transmission speed	25 kBps	25 kBps
Transmission range	10 m, 30 m	100 m
Node speed	0.8 m/s . . . 2.0 m/s	5 km/h . . . 50 km/h
Message size	100 B–200 B	100 B–200 B
Message creation interval	[60–180] s	[60–180] s
Buffer size	10 kB, 100 kB	100 kB, 1 MB
Number of nodes	10, 20, 40, 70	320

The metrics used to compare the performance of the protocols are described in the following:

- *Delivery ratio* is the ratio between the delivered messages over the generated ones.
- *Overhead* is the ratio between the total number of transmissions over the number of delivered messages.
- *Average delay* is the average time needed for a message to reach its destination.

Those metrics have been used not only to evaluate the performance of the DTN protocols, but also to evaluate to what extent the proposed techniques, namely QM, FW and R, impact performance.

The results are obtained varying the above parameters for each protocol and averaging the results over fifteen runs. Each experiment was repeated as many times as was deemed appropriate in order for the standard deviation of the results to lie within the 95% confidence interval.

We note that the existing version of the ONE simulator implements only one of the above mobility models (namely, random walk). We extended ONE by implementing the two other synthetic mobility models (Levy Walk [16] and Community [6,22]) and a mechanism for reproducing the motion of nodes based on real traces of mobility [23,24]. Our implementation of the two models and the extensions to reproduce real-world mobility traces are open-source and available online, to be used freely by other researchers [27].

3.2. Evaluation Using Synthetic Mobility Patterns

We here discuss the experimental results obtained simulating 10 of the protocols discussed in Section 2. We categorize the protocols into three classes of protocols: simple, single technique and advanced protocols. For each class, we discuss how and why the proposed techniques affect the performance metrics. In order to conduct the comparative study, we implemented three DTN routing protocols, namely Fuzzy-Spray [25], FAD [6] and SCAR [26]. Our protocol implementations are available online along with the other extensions made to the ONE simulator.

We here include the results for the random walk mobility model and the Levy Walk mobility model (see Figures 2–4). The results related to the Community model were not included in full detail; instead, they are presented in comparison to the other two models in Section 3.4.

3.2.1. Simple Protocols

We use the term "simple protocols" to refer to DTN routing protocols that do not use knowledge information in managing the messages. They are among the first protocols proposed in the literature. Since they avoid collecting information on the current status of the network, they are less demanding protocols in terms of computational capabilities.

Epidemic [35] explores all possible options, namely all nodes are possible carriers. This implies that when buffer constraints are negligible (see 100 KB in Figure 2), it is the best performer in terms of delivery ratio and delay. On the contrary, when the size of the buffer becomes critical (10 KB), as the density of the network increases, the delivery ratio quickly decreases due to message dropping. As a consequence, a number of messages need to be sent again, and the overhead increases. We stress that this result makes clear the importance of an effective queue management policy when the nodes' buffer size is small.

First-Contact and Direct-Delivery [20] show the same performance irrespective of the buffer size, which is in any case sufficient to store all of the messages in the network at any given time. Essentially, the only difference in the approach of the two protocols is the forwarding policy, yet it does not have any impact on delivery ratio and delay. We only observe (by taking into account possible error, our performance evaluation is inherently prone to) that First-Contact involves a larger overhead of messages, as it forwards the message to the first encountered node until the destination is reached.

In the following sections, we will evaluate the knowledge-based forwarding policies and discuss how they can help improve the performance also when limited replication is employed.

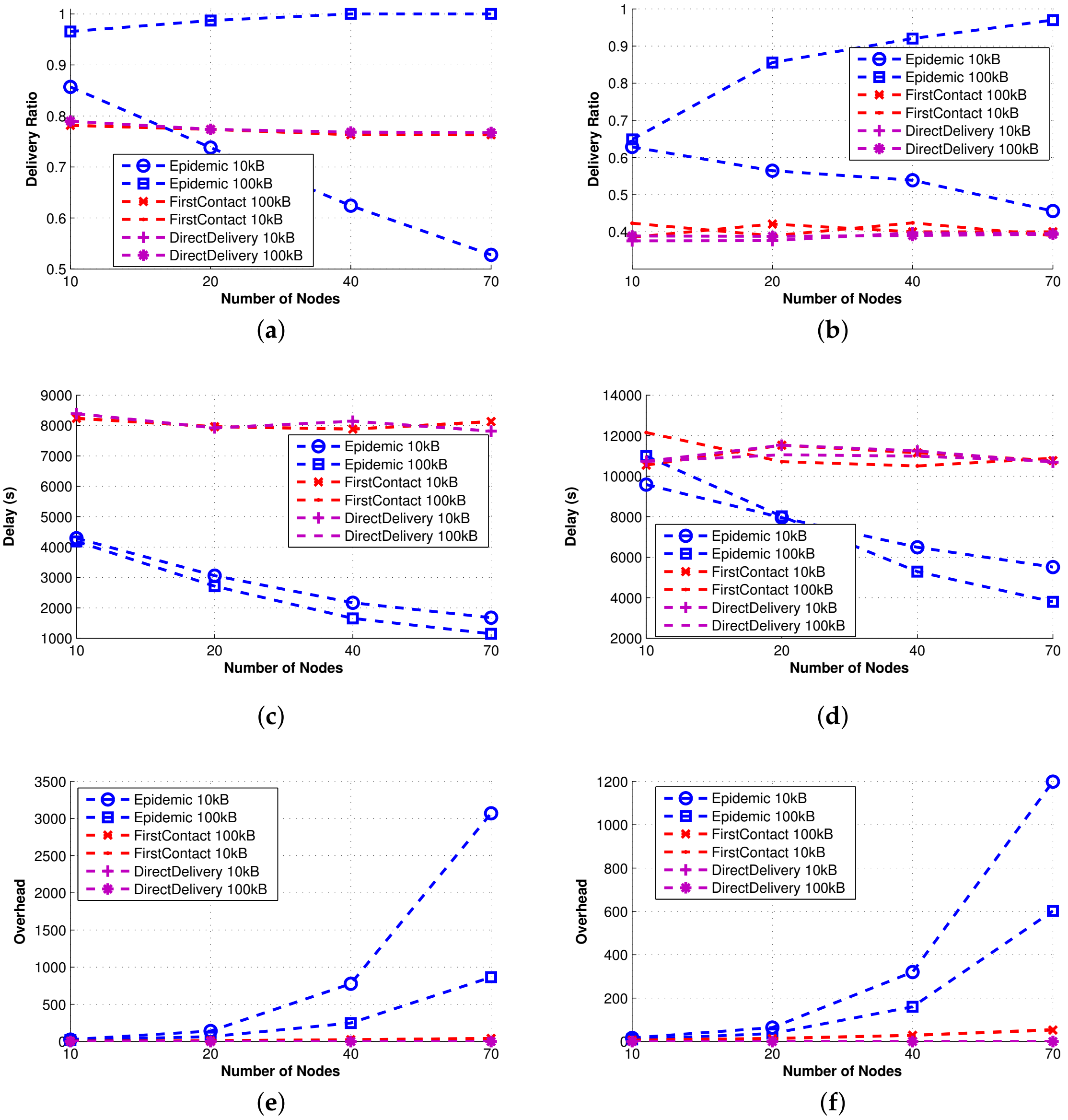


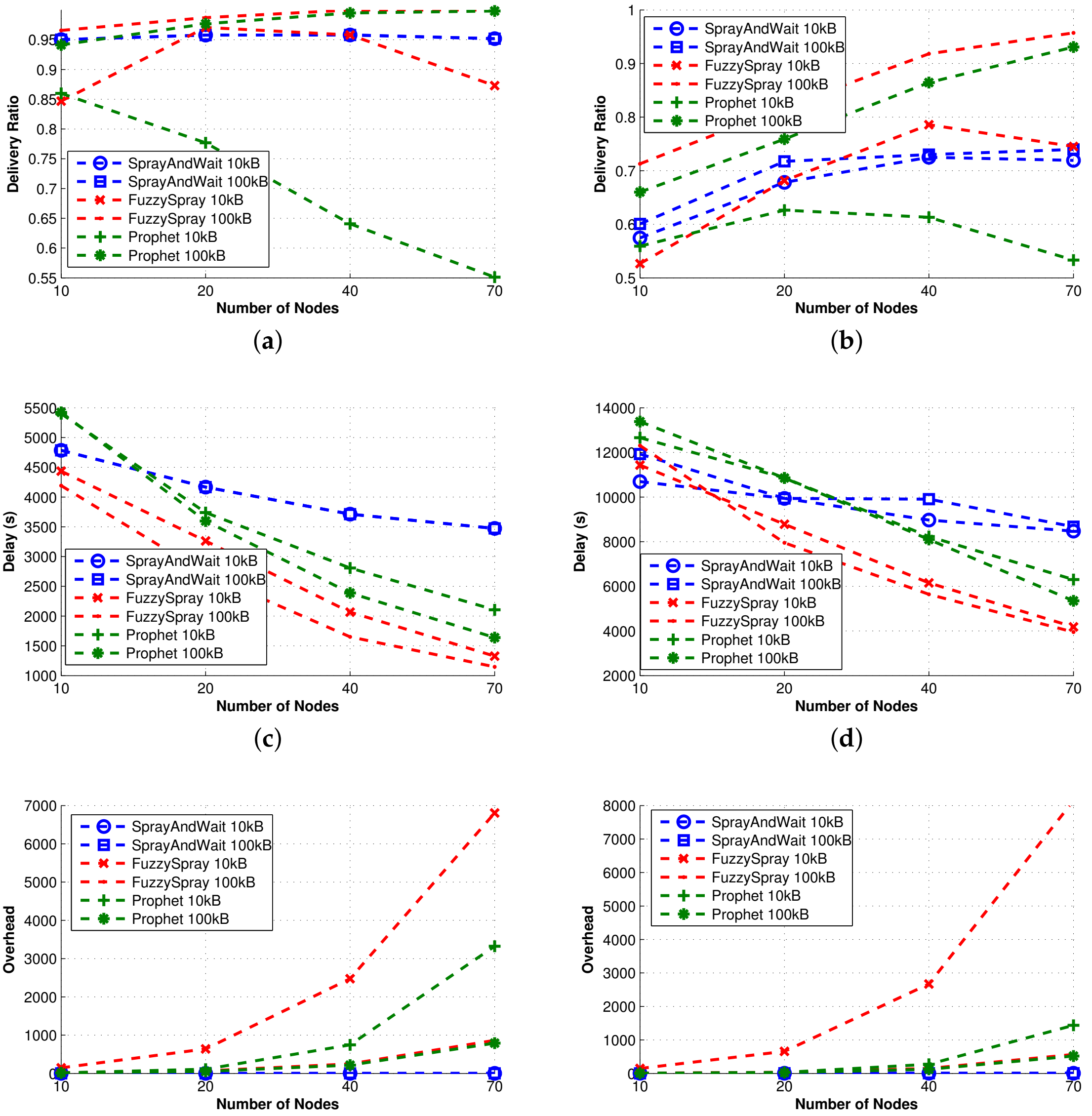
Figure 2. Synthetic traces. Random walk (**a**,**c**,**e**); Levy Walk (**b**,**d**,**f**). Simple protocols.

3.2.2. Single Technique Protocols

We now examine how simple protocols can be improved using just one of the proposed techniques (see Figure 3). As such, the protocols analyzed in this section implement either QM or FW or R.

Spray-and-Wait [36] implements the replication technique. It shows good performance apart from the delivery delay. The average overhead is bounded by the number of copies created in the sprayphase. Based on [36], the L parameter for Spray and Wait is recommended to be chosen for 10%–15% of the expected number of nodes. Therefore, given the network size, the node mobility patterns and application message generation rates, we set set $L = 4$, so that on average, three copies of the messages are forwarded in the spray phase. In the waitphase, those copies have to reach the destination in a direct delivery fashion. For this reason, the delay of Spray-and-Wait is comparable to that of Direct-Delivery; it is always smaller, because the multiple copies of the message in the network increase the probability of reaching the destination in less time.

PRoPHET [37] implements the forwarding technique. When the number of nodes increases, the forwarding technique has more candidates as possible effective relays among the contacts. This implies that the overhead increases (more relays imply more transmissions) and the delay decreases (more copies of the message increase the probability of reaching the destination). However, the effective management of the multiple copies of the message requires either a proper queue management technique or a big buffer size. In fact, for the small size of buffer considered, the delivery ratio of PRoPHET decreases due to message dropping.



(**a**)

(**b**)

(**c**)

(**d**)

(**e**)

(**f**)

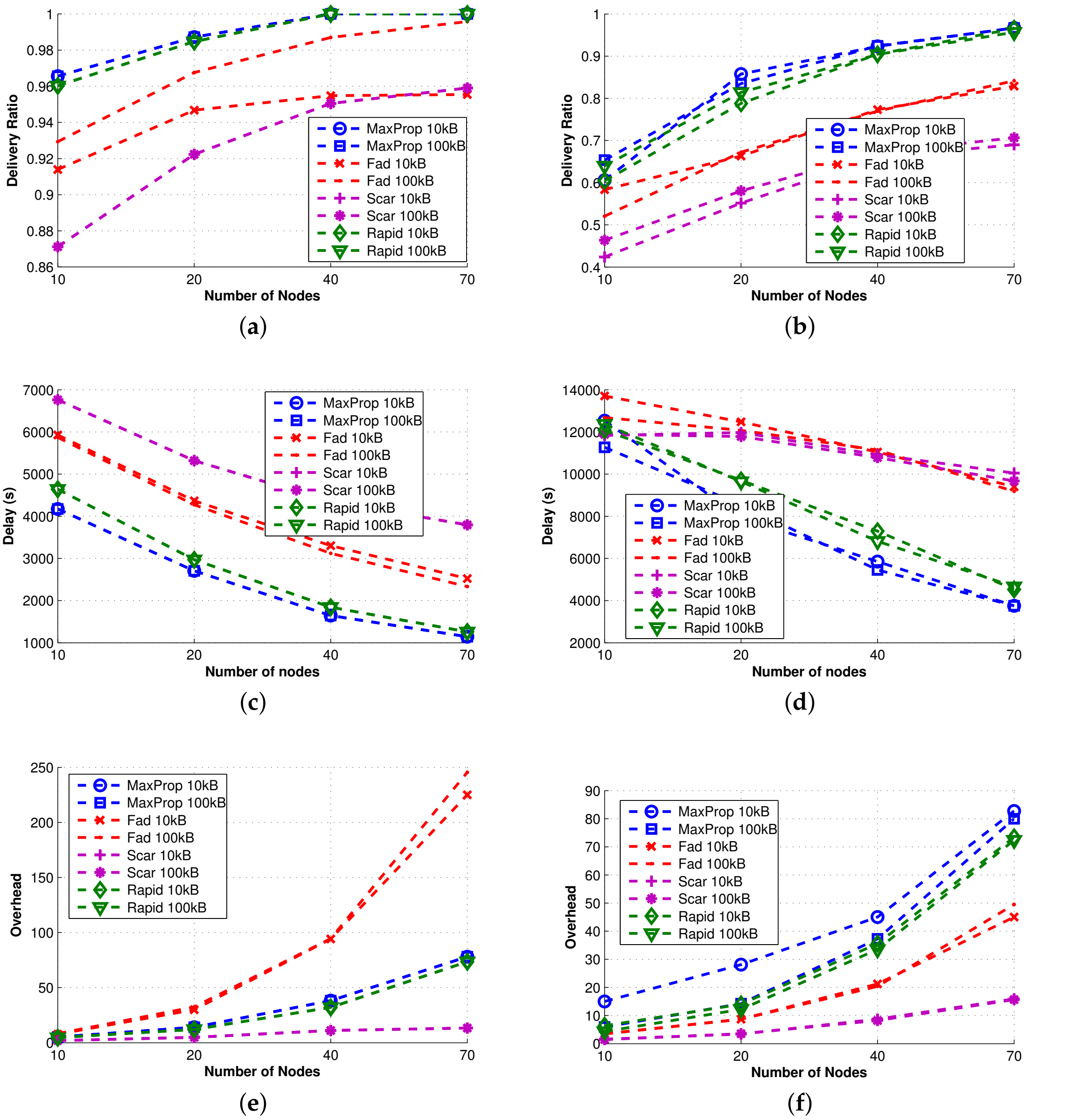
Figure 3. Synthetic traces. Random walk (**a**,**c**,**e**); Levy Walk (**b**,**d**,**f**). Single technique protocols.

Fuzzy-Spray [25] implements the queue management technique. It is the best performer in terms of delivery delay, because (similarly to epidemic) there is an unlimited number of replicas in the network. QM tries to keep messages in the queue until there is a high confidence that they have been delivered already. As a consequence, even when the size of the queue is relatively small, the effects of buffer overflow and message dropping are less dramatic (note the differences with Epidemic). Indeed, the delivery ratio is always remarkably high, except for the highest density of nodes (70 nodes in Figure 3);

this is because, in such a scenario, the QM policy employed by Fuzzy-Spray does not effectively handle all messages.

3.2.3. Advanced Protocol

We proceed by examining advanced protocols that combine more than one technique. The effectiveness of these approaches are shown in Figure 4, where the delivery ratio is always remarkably high. Furthermore, the overhead grows almost linearly (except for FAD), and it is in any case an order of magnitude smaller than that shown by single technique protocols. To our understanding, this experimental evidence indicates that the combination of the proposed techniques can significantly increase the scalability of the protocols.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 4. Synthetic traces. Random walk (**a**,**c**,**e**); Levy Walk (**b**,**d**,**f**). Advanced protocols.

FAD [6] implements both queue management and the forwarding techniques. Both depend on the estimation of the message destination meeting probability; however, the accuracy of this value is highly coupled to the movement patterns among the nodes. Thus, the combination of these two techniques results in a good performance using levywalkmobility (or real traces, as we will see later), while in a very high overhead using random walk mobility (see Figure 4e,f). Although the designers of FAD attempted to limit the forwarding, as the main thing responsible for overhead, examining its behavior in a broader set of protocols, but under a dedicated community-based movement, we understand that their design goal does not achieve high success under random movements of nodes. Another possible explanation of this behavior is that (1) in the considered scenario, TTL is set to infinity and (2) FAD keeps in queue the most important messages, namely the messages with lower fault tolerance values. Those messages are never evicted (TTL is infinity) and are often retransmitted, increasing the overhead. To confirm this fact, we conducted some other experiments where the TTL is set to a limited value. In this additional set of experiments, FAD experiences a low overhead.

MaxProp [29] and RAPID [18] are two advanced protocols that implement queue management and forwarding techniques. In contrast to FAD, their forwarding policy introduces message reordering rules, but without filtering out any message. Despite the different forwarding approach, the performance achieved is very similar. When a filter is applied, messages can be forwarded or not, while re-ordering implies that all messages can be possibly forwarded according to the new ordering, if there is a sufficient amount of time. The QM technique guarantees a high delivery ratio (even with a small buffer size) and a low delay, as well as limited overhead, even if unlimited replication is employed.

SCAR [26] is the last protocol considered in this section and is the only protocol implementing all of the proposed techniques. However, the SCAR queue management is very simple (it only uses master-backup copies), and in the considered scenario, the context information (*i.e.*, battery level, mobility pattern, *etc.*) is similar for all nodes, limiting the effectiveness of the forwarding. As a consequence, SCAR shows high delay and a relatively small delivery ratio, while the overhead is always small thanks to the effectiveness of the replication technique.

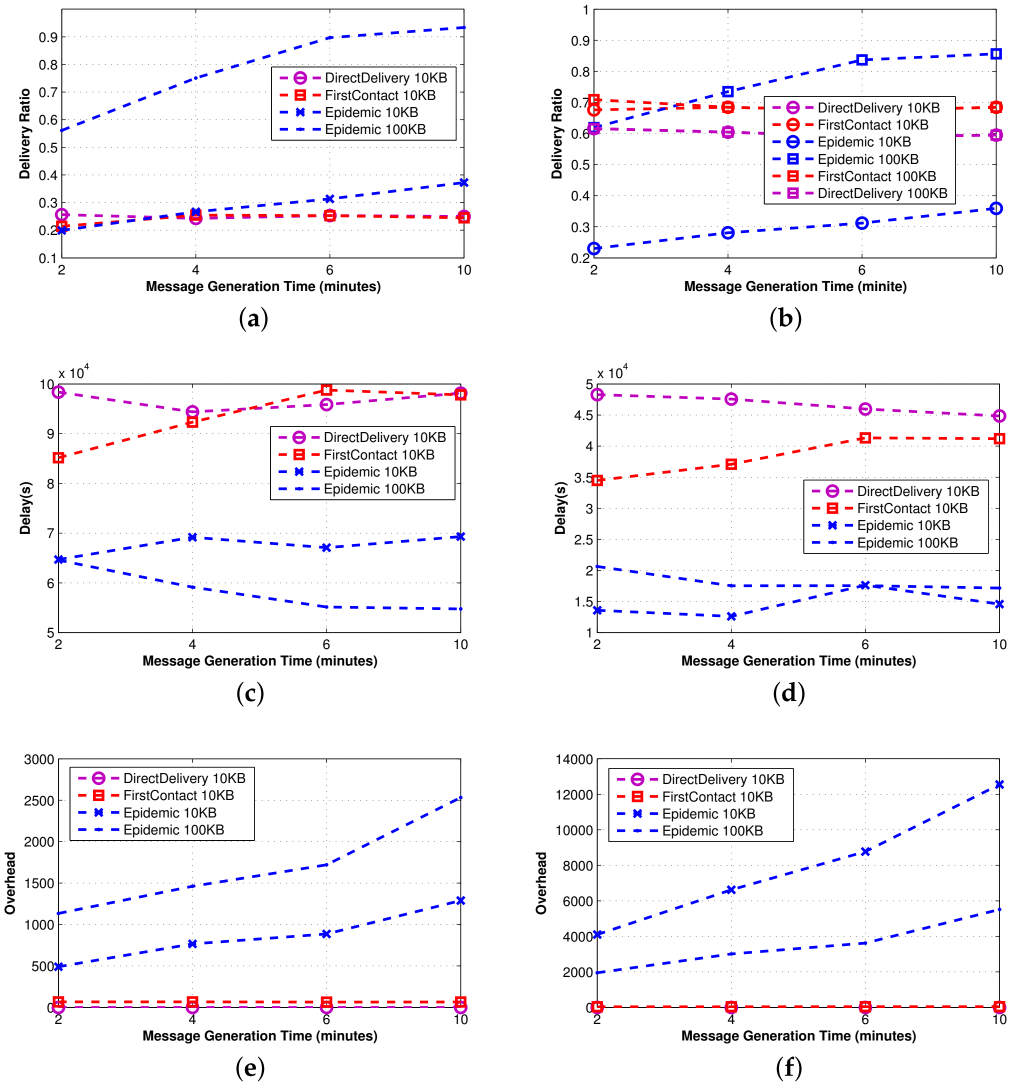
3.3. Evaluation Using Real-World Mobility Traces

We continue by evaluating the performance of the protocols by conducting an experiment using real-world traces. The motion of the nodes is dictated either by the traces acquired by the Haggle experiment [23] or by the traces acquired by the Rome Taxi experiment [24]. Independent of the movement, at any given time, a node that decides to generate a new message selects a random destination and invokes the underlying DTN routing protocol. Messages never expire, namely their TTL is infinity. The simulation settings are summarized in Table 3. It is worth noting that metrics on synthetic traces are obtained averaging the results over fifteen runs in which the underlying connection graph is different in each run. This is clearly not possible for real traces, in which the underlying connection graph is obtained from the traces, as explained earlier.

Table 3. Simulation settings.

Parameter	Indoor Environment	Outdoor Environment
Simulation time	4 days	30 days
Transmission speed	25 kBps	25 kBps
Transmission range	30 m	100 m
Message size	100 B . . . 200 B	100 B . . . 200 B
Message creation interval	2, 4, 6, 10 (min)	5, 10, 15 (min)
Buffer size	10,100 (KB)	100 KB, 1 MB
Number of Nodes	78	320

In Figures 5–7, we include the results for the real traces and the Levy Walk mobility model. We decided to include the results using the Levy Walk as a reference when comparing the results with those of the previous section. A more in-depth comparison of the results acquired from each model is presented in Section 3.4.

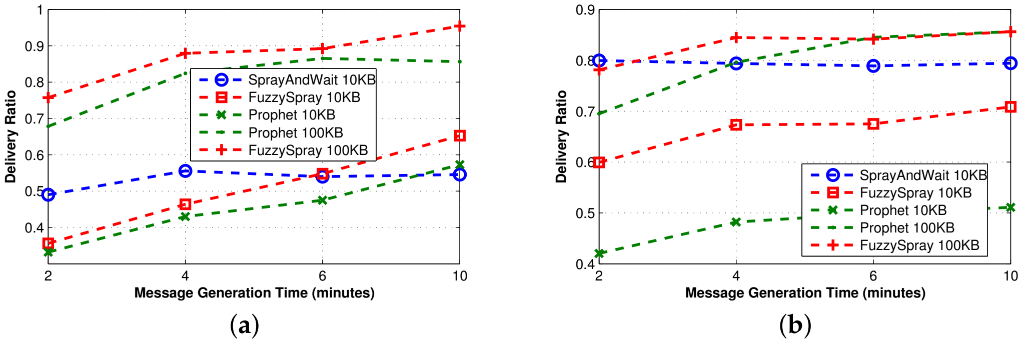


(**a**) (**b**)

(**c**) (**d**)

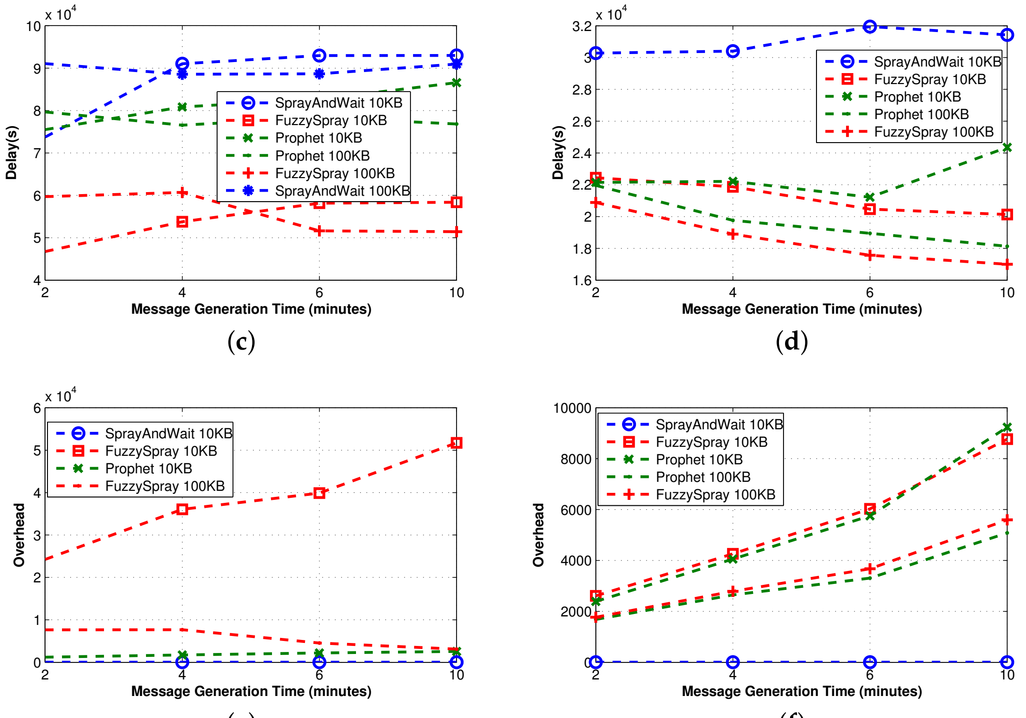
(**e**) (**f**)

Figure 5. Simple protocols: Levy Walk (LW) (**a**,**c**,**e**) *vs.* Infocom (**b**,**d**,**f**). For the sake of readability: if the difference between the 100-KB and 10-KB plots for a protocol is within 5%, we do not show the 100-KB plot.



(**a**) (**b**)

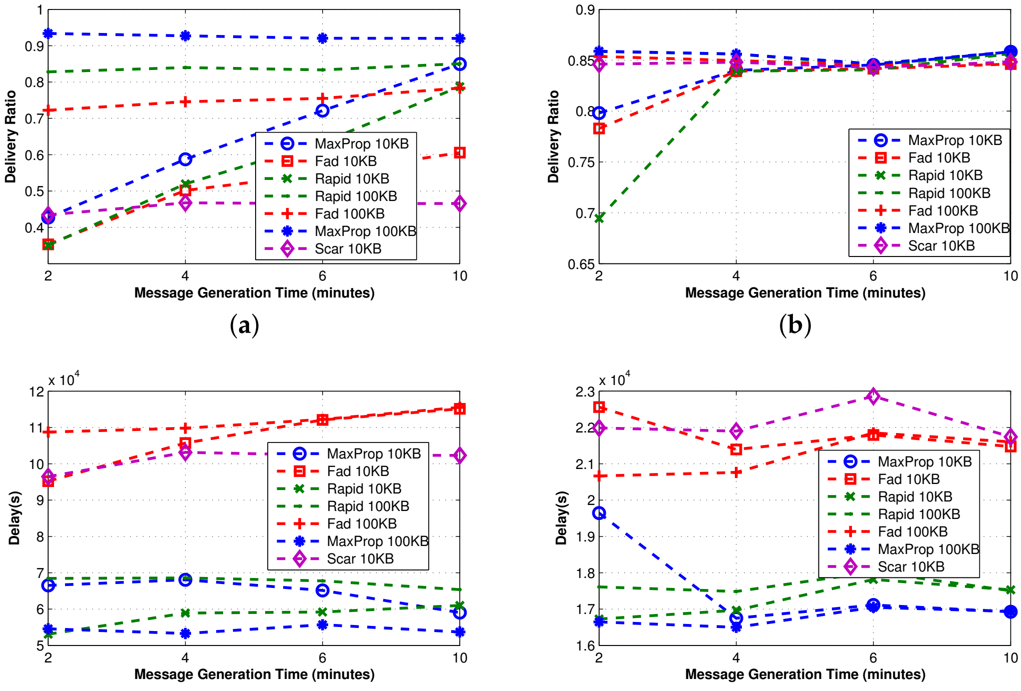
Figure 6. *Cont.*



(**c**) (**d**)

(**e**) (**f**)

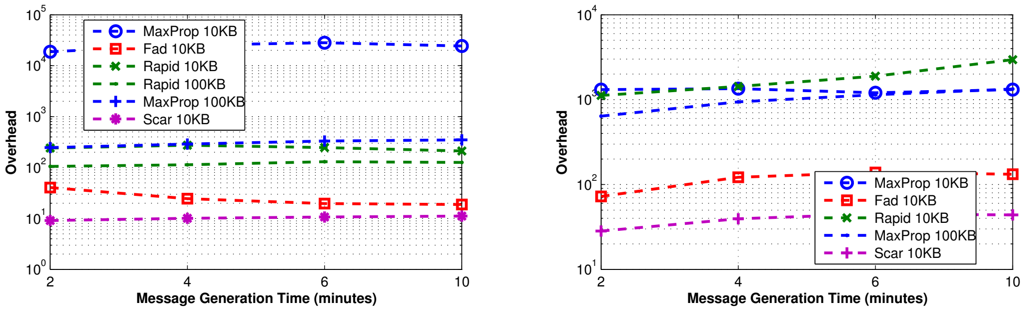
Figure 6. Single technique protocols: LW (**a**,**c**,**e**) *vs.* Infocom (**b**,**d**,**f**). For the sake of readability: if the difference between the 100-KB and 10-KB cases for a protocol is within 5%, we do not show the 100-KB case.



(**a**) (**b**)

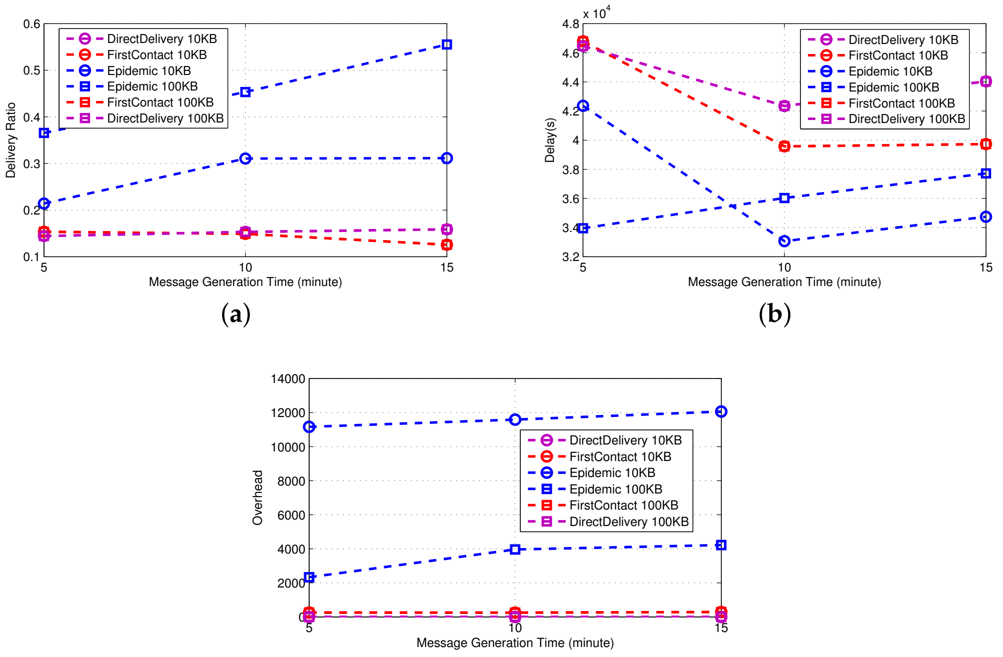
(**c**) (**d**)

Figure 7. *Cont.*



(**e**) (**f**)

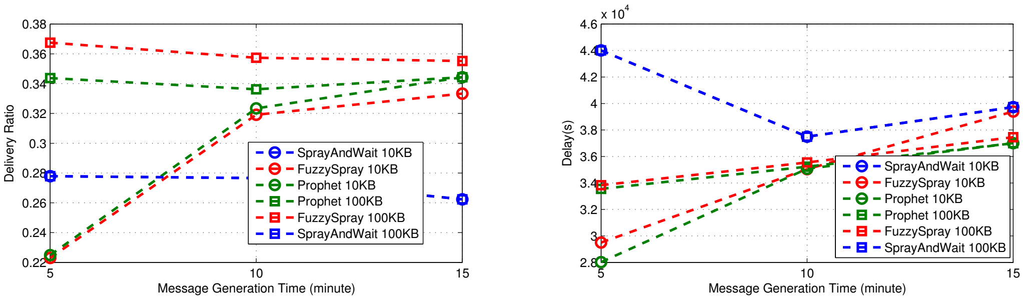
Figure 7. Advanced protocols: LW (**a**,**c**,**e**) *vs.* Infocom (**b**,**d**,**f**). For the sake of readability: if the difference between the 100-KB and 10-KB cases for a protocol is within 5%, we delete the 100-KB case.



(**a**) (**b**)

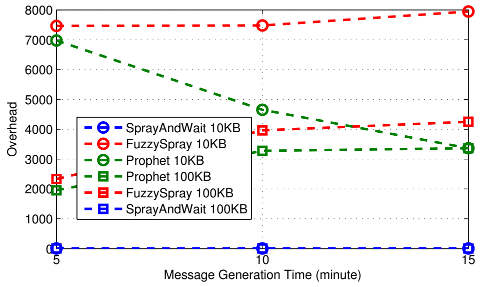
(**c**)

Figure 8. Performance of simple protocols in a real-world outdoor setting.



(**a**) (**b**)

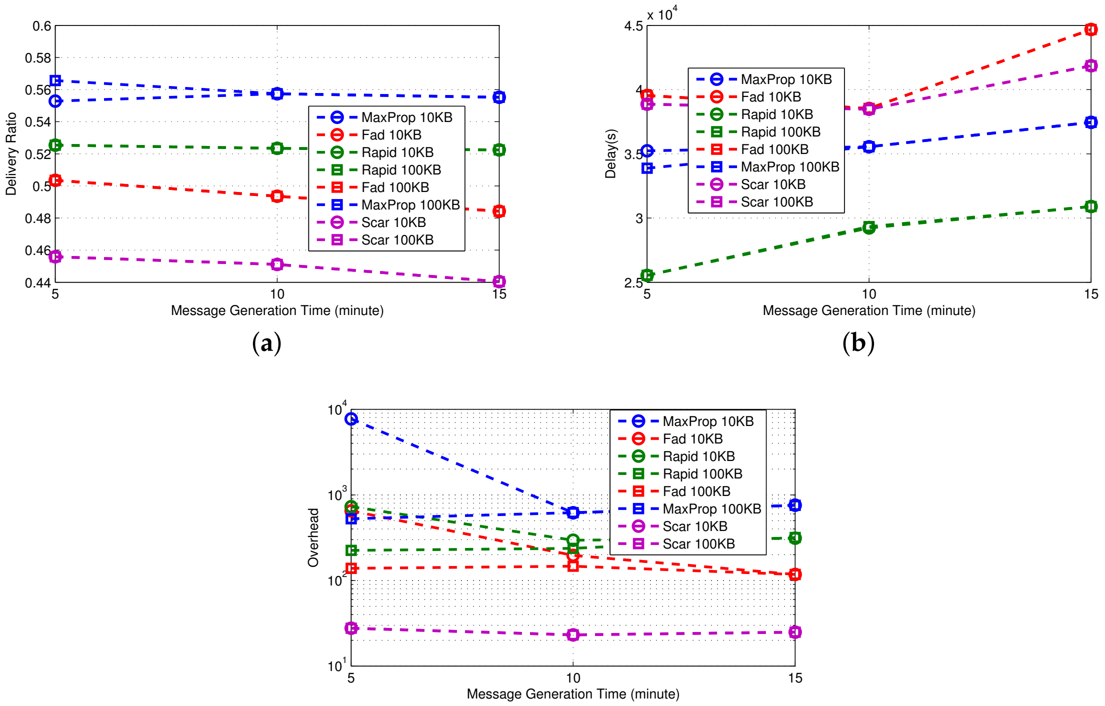
Figure 9. *Cont.*



(**c**)

Figure 9. Performance of single technique protocols in a real-world outdoor setting.

In Figures 8–10, we include the results for the real traces based on the Rome Taxi experiment. The outdoor traces cannot be compared to the Levy Walk mobility model or any other synthetic mobility model, since all of them are used to model the motion of human walking, while the real-world trace is based on the mobility of cars.



(**a**) (**b**)

(**c**)

Figure 10. Performance of advanced protocols in a real-world outdoor setting.

3.3.1. Simple Protocols

The main parameter that affects the performance of simple protocols is the meeting opportunity, namely the probability of meeting another node. An indirect indication of the meeting opportunity is given by observing the performance of Direct-Delivery and recalling that in this protocol, a message is directly delivered to the intended destination. As a consequence, the observed metrics only depend on the source-destination meeting opportunity.

We observe that although the Inter-Contact Time distribution (ICT) is similar for the real traces and the synthetic model (see Figure 5), the meeting opportunity on real traces is higher than the one in synthetic traces. Indeed, the delivery ratio of Direct-Delivery is higher (Figure 5b *vs.* 5a), and the delay is lower (Figure 5d *vs.* 5c). Similar arguments apply also to First-Contact, where a single copy of the message is forwarded in a multi-hop fashion.

The higher meeting opportunity of real traces is confirmed by the fact that the overhead generated by Epidemic is significantly higher for real traces. However, a higher overhead does not provide a higher delivery ratio. In Epidemic, the message is spread to all of the encountered nodes; even if in the synthetic case the meeting opportunity is lower, the number of copies of the messages is sufficient to achieve a delivery ratio comparable to the one observed in real traces. Clearly, the delivery ratio is higher when a bigger buffer is used.

Regarding the outdoor mobility traces, by examining the performance of the Direct-Delivery protocol, it is evident that the network is much sparser than the indoor cases, and the inter-contact time is much longer. The delivery ratio of the Direct-Delivery protocol is as low as 10% throughout the execution of the experiment. In [48], the mobility trace is analyzed using statistical techniques, graph theory and network analysis. Their analysis indicates that the nodes are forming groups based on their mobility patterns, that is the group of vehicles that share many common encounters, but not meeting often. In this sense, the connectivity graph is partitioned and remains partitioned during the execution of the experiment, thus making direct delivery impossible.

As indicated in [5], it is therefore evident that the only approach to improve the performance of the network is to relay messages through the other nodes of the network. As we observe, the Epidemic protocol achieves much higher success rates that go above 50% as the message is spread to all of the encountered nodes, and eventually, one copy reaches the destination. Clearly, this approach speeds-up the delivery of the nodes (*i.e.*, reduces the delivery delay), while as each message is copied multiple times, the overhead of the protocol is also higher.

3.3.2. Single Technique

In Figure 6, we observe that Fuzzy-Spray can be roughly described as Epidemic implementing also a QM technique. For this reason, its behavior is similar to the one observed in Epidemic, but the effectiveness of the QM allows Fuzzy-Spray to obtain significantly better delivery ratios; this is particularly evident in the small buffer case. The effectiveness of the QM with small buffers is confirmed comparing Fuzzy-Spray to PRoPHET and recalling that PRoPHET implements only the FW technique. Spray-and-Wait generates a limited number of copies (four in our experiments). When the meeting opportunities are relatively high, as in real traces, this number of copies allows Spray-and-Wait to provide a fairly good delivery ratio. On the contrary, in synthetic traces, the meeting probability is lower, and thus, such a number of copies cannot provide the good performance observed in real traces.

The high number of copies generated by Fuzzy-Spray increases the probability of a fast delivery; indeed, the delivery delay achieved by this protocol is the lowest achieved among the 10 protocols considered, but at the same time increases the overhead. Similarly, the limited number of copies generated by Spray-and-Wait results in a higher delay and in a lower overhead. The effectiveness of the QM is more evident when the meeting opportunities are lower, as in the synthetic traces; while for real traces, the performance in terms of delay is pretty similar and mainly depends on the buffer size.

In terms of the outdoor real-world mobility trace, the performance of the single technique protocols is roughly the same. As with the case of the simple protocols, the nature of the mobility of taxis in the metropolitan area heavily affects the overall performance of the protocols, thus keeping the delivery ratios much lower than in the case of the indoor real-world mobility trace. Therefore, since the meeting opportunities remain very low, Spray-and-Wait achieves the lowest delivery ratio, while the performance of the other two protocols is almost identical.

3.3.3. Advanced Protocols

In Figure 7, we first observe that in real traces, all of the protocols behave similarly in terms of delivery ratio, when message generation time is sufficiently high. This means that the advanced techniques used in the protocols are comparable for this metric, if the frequency of new messages injected into the network is not too high. However, the delay of RAPID and MaxProp is remarkably lower, and this is also confirmed on synthetic traces. The price to pay for such better performance is overhead, but it is worth noting that MaxProp with 100 KB of buffer outperforms the other protocols in terms of overhead and at the same time provides excellent performance in terms of delivery delay and delivery ratio. The reason for this is that the size of the buffer is sufficient to "permanently" store all of the relevant messages. As a consequence, such messages do not have to be retransmitted, and this results in a lower overhead.

We also observe that the behavior of the protocols on synthetic traces is similar to the one observed in real ones, with one significant difference: the meeting opportunities. As an example, SCAR on real traces performs reasonably well in terms of delivery ratio, irrespective of the message generation time, while it achieves the worst performance on synthetic traces. We understand that, even if SCAR implements FW and QM, it limits the number of copies of a message in the network. As a consequence, only the nodes (four in our settings) having a copy of the message can deliver it to the destination, and thus, this is the protocol that is more influenced by the meeting opportunities, which are lower in the synthetic traces. This is also confirmed by the fact that SCAR, apart from MaxProp 100 KB, is the protocol with the lower overhead.

Once again, the observations on the operation of the protocols based on the indoor case are comparable to the ones obtained for the outdoor case. In the outdoor case, the performance of the three protocols is comparable, with MaxProp achieving a slightly higher delivery rate compared to the other two protocols. As before, the delay of RAPID and MaxProp is lower than the other protocols. Similarly, the overheads of the protocols are similar to the indoor case.

3.4. Impact of the Mobility Model on Performance

Mobility models clearly affect nodes' opportunity to meet other nodes, namely the probability to reach the destination and the inter-contact probability. Consequently, also messages' likelihood to be delivered before being dropped depends on mobility models. In this section, we evaluate the impact of the mobility models on some of the protocols we discussed in the previous sections. Moreover, the relevance of the information exchanged between nodes to update their knowledge on the environment in many cases strongly depends on mobility models. As an example, social information in RWP is useless, while the same information in the CM is used to predict the expected movement pattern of nodes.

Figures 11 and 12 show the effects of different mobility models on the delivery ratio and delay of some of the protocols discussed earlier. When the movement pattern is controlled by some intentional behavior (as in LW and CM), protocols using the knowledge (e.g., MaxProp) perform better, and in any case, the gap between such protocols and the ones that do not use the knowledge is remarkably high. When random movement patterns (as in RW, RWP) are considered, this gap becomes less evident, due to the fact that the knowledge acquired by nodes can be hardly used to predict future behaviors.

Surprisingly, the delivery ratio of knowledge-based protocols is higher in the experiments based on random mobility models. As an example, the delivery ratio of MaxProp when 10 nodes are considered is more than 98% in Figure 4, while it is about 92% in Figure 11. This behavior can be explained by the fact that in the random mobility models, the probability of meeting other nodes is significantly higher and more uniform than in LW and CM. The evidence of this is that the Direct-Delivery protocol, which only delivers messages to the destination, has a delivery ratio that is very high (more than 75% for all densities) in the random mobility case (see Figure 2) and relatively smaller for LW an CM (see Figure 11). This fact eases the delivery of the messages to random selected destinations. In any case, we stress again that the gap between knowledge-based protocols and other protocols is pretty evident when LW and CM are used.

We finally compare the results of the experiments performed on real-world traces to those of the experiments using synthetic traces (the simulated area is one square km) generated according to the Levy Walk (LW) mobility model, in which the Inter-Contact Time (ICT) distribution, *i.e.*, the time period between two consecutive contact times of the same two nodes, approximates the one observed in real traces. The ICT reflects the mobility pattern of the nodes. In [16], it is shown that the ICT distribution of human walking exhibits a power-law tendency up to some time after which it shows exponential decay. The authors show that adjusting the LW parameters, it is possible to generate an ICT distribution similar to the one observed in Infocom traces.

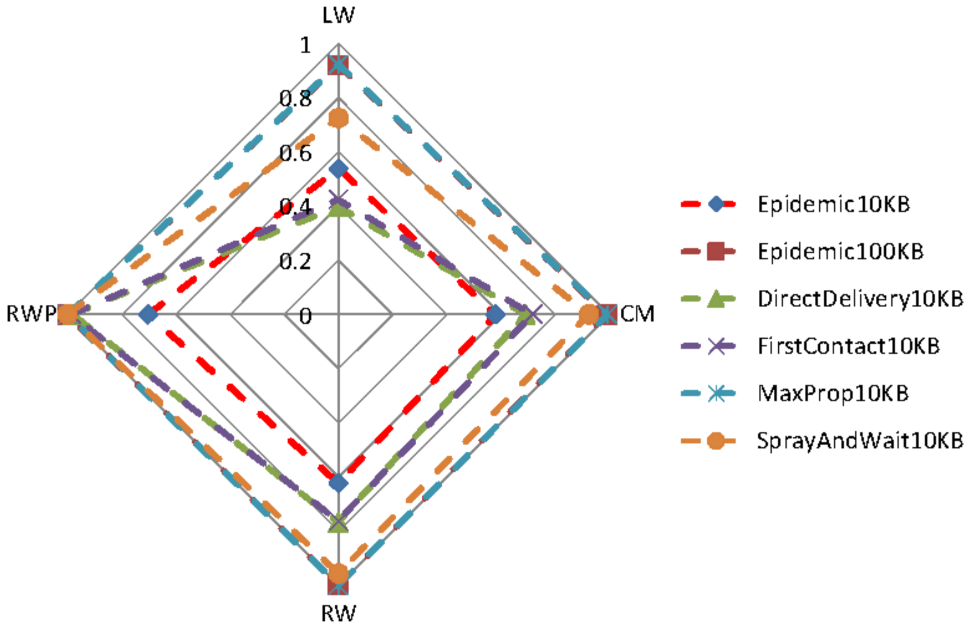


Figure 11. Delivery ratio with different mobility models, 40 nodes.

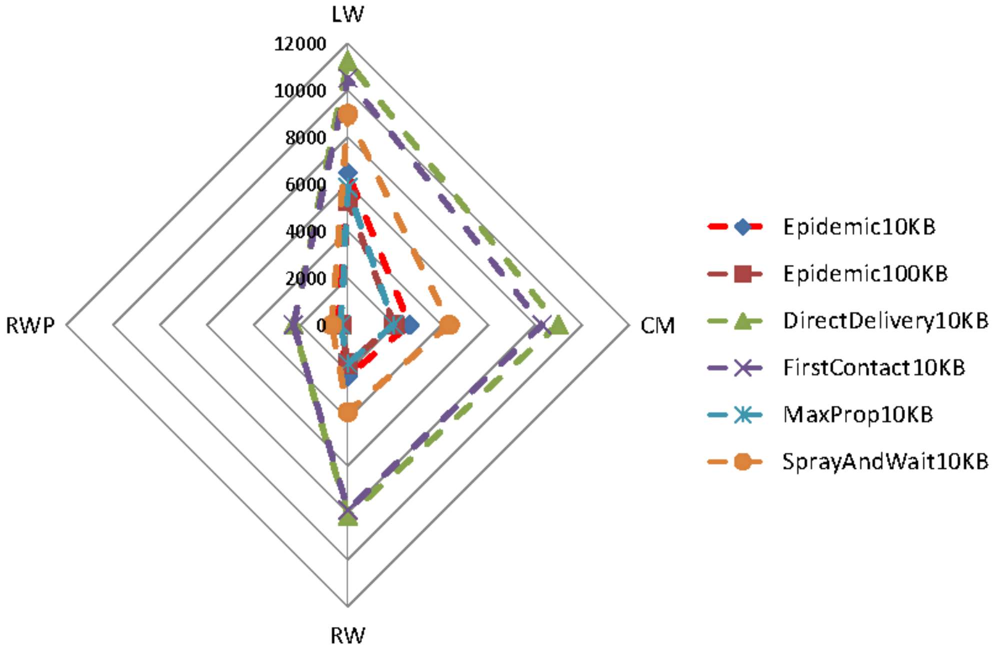


Figure 12. Delay with different mobility models, 40 nodes.

Figure 13 shows the similarity between the ICT distribution extracted from the Haggle traces and the one obtained from LW, setting the parameters as suggested in [16]. From this figure, it is clear that though the two distributions are similar, in real traces, we expect more nodes having longer ICT than in synthetic ones.

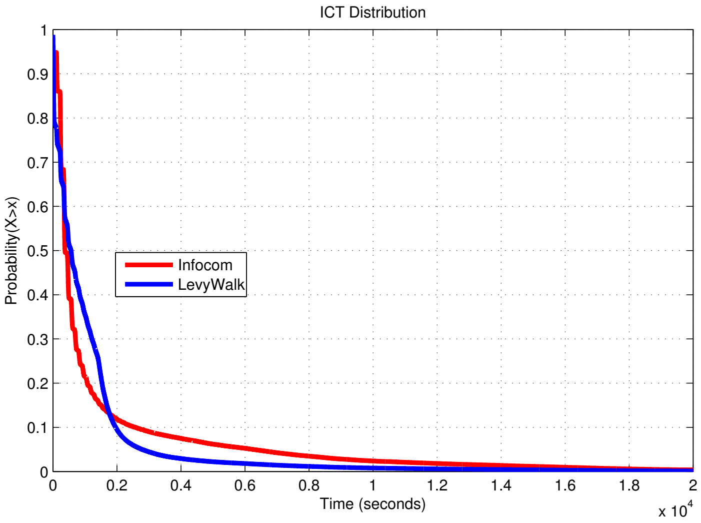


Figure 13. ICT distribution for Infocom traces and Levy Walk in our scenario.

For the case of the outdoor mobility trace, it is evident that the much larger network area affects the overall performance of the protocols, as it takes much longer for nodes to meet. In addition, the fact that some nodes do not meet each other and there exist periods of time where the network is partitioned, the overall delivery ratio is much lower than that observed for the indoor case. However, since the mobility of the nodes cannot be affected by the protocols, it is impossible to overcome this problem and design a protocol that achieves higher delivery rates or reduces the delivery delay. One should follow a different approach, for example using the approach of [5], where the mobility patterns of a very small subset of the nodes is dictated by the communication protocol in order to overcome cases of network partitions.

4. Technical Recommendations

In the previous section, we examined 10 well-established DTN routing protocols and comparatively evaluated their performance. We looked into their mechanisms and how they are related to the three basic components used by the reference architecture proposed. In this section, we summarize the discussion by establishing a connection across the three core components, the approaches used to implement them in the group of protocols studied and the most important characteristics of DTNs. Our objective is to provide technical recommendations and design considerations for protocol engineers and network administrators that can take into account when selecting or designing new routing protocols, based on the given set of application characteristics and requirements. Table 4 provides a synopsis of the technical recommendations.

The main properties of DTNs that should be considered when designing a DTN routing protocol are: (1) contact time and node mobility patterns; (2) node resources; and (3) network traffic imposed by the application.

The FW technique is important when nodes' movement patterns are highly correlated. Forwarding messages to nodes that will likely meet the final destination more frequently improves the delivery ratio and reduces the delivery delay, and in particular, when contact time is critical (*i.e.*, it is short), the selection of appropriate messages to be forwarded is crucial. On the other hand, as movement patterns become less correlated, the importance of FW is reduced; essentially, the acquired knowledge does not help predict the future. An important element to consider in the set-up of forwarding filters, rules and policies is the availability of memory to buffer the messages. Excessive forwarding of messages will generate message buffer overflows, thus eventually hindering the overall performance.

In all practical cases in which the buffer size is limited, irrespective of the mobility patterns, QM should be well designed taking into consideration the expected traffic and the implications of the adopted FW and R techniques. Clearly, more traffic is generated, either by the application itself or by aggressive FW and R techniques, and more load is handled by the QM.

When designing a QM component, it is important to provide rules and mechanisms for prioritizing messages according to their destination. This can be exploited if nodes have predictable movements. It is equally important to also prioritize messages in terms of age, hops or number or replicas in the network. Another very important parameter for tuning the performance of the QM component is message TTL. Infinite TTL should be avoided, unless the application is generating a low volume of messages.

QM highly affects the performance of the protocols. For example, we have seen that thanks to an effective QM policy, MaxProp shows good performance in terms of delivery ratio and delay, even if a high number of messages is dropped, primarily due to the unlimited replicas generated by the protocol.

Finally, the replication technique is necessary to control the number of message copies. The larger the number of replicas, the better the chance to reach the destination, but the more network resource bandwidth and buffer space are needed. For example, when contact opportunities and mobility are low, but the nodes have enough resources in terms of bandwidth and buffering, more replication will increase reliability. On the other hand, if the nodes are highly mobile and nodes' resources are limited, less replication would better utilize the bandwidth.

Table 4. Synopsis of technical recommendations.

System Property	Recommendation
Predictable Movement Patterns	Prioritization of messages in terms of age, hops and number of replicas.
Correlated Movement Patterns	Combination of forwarding techniques and queue management (e.g., FAD) improves overall network performance.
Uncorrelated Movement Patterns	Direct-Delivery keeps network overhead at minimum levels.
Sparse Contacting	Increased message replication increases reliability. If there is large availability of message storage, Epidemic-based techniques achieve high performance.
High Node Mobility	Limited message replication improves bandwidth utilization. Forwarding techniques (e.g., PRoPHET) reduce latency. Queue management techniques (e.g., Fuzzy-Spray) do not affect performance.
Low Message Generation Rate	Setting high values of message TTL improves performance.
Low Message Storage Capacity	Forwarding techniques (e.g., PRoPHET) hinder performance due to increased message drop rates. Queue management techniques (e.g., Fuzzy-Spray) maximize performance.

5. Conclusions and Future Work

We identified forwarding, queue management and replication as the core components for designing efficient and effective DTN routing protocols. On the basis of these components, we proposed a *reference architecture* and discussed how a large group of existing and well-established DTN routing protocols presented in the relevant literature can be used to implement it. Contrary to most previous papers, where either only qualitative comparisons have been presented or only a single category of protocols have been analyzed, in this work, we conduct an extensive comparative study of the performance of 10 DTN routing protocols. To the best of our knowledge, this is the first time that such a large set of protocols has been examined. The key criterion for selecting these protocols is the techniques employed given the three core components of our reference architecture. We suitably selected protocols such that we can understand the effect of each component in stand-alone and in combination with the others. Therefore, the performance evaluation conducted goes beyond pair-wise comparison of protocols that either follow similar techniques or are completely different, since it can provide insights on the prevailing design choices and crucial protocol properties. Our results show that an effective combination of the proposed techniques can significantly improve the performance of the protocols in terms of delivery ratio, overhead and delivery delay.

In our extensive study, we used a combination of synthetic mobility models and models based on traces acquired from monitoring the actual motion patterns of nodes when executing real-world application. Our evaluation was done based on the broadly-accepted ONE simulator. We have extended the tool to include two additional synthetic mobility patterns and a generic mechanism to replay traces from real-world experiments. Based on the results of the synthetic and realistic traces, we provide a comparison between the two approaches and identify how and when each model should be employed during the evaluation process of a DTN routing protocol.

Our implemented models and extensions along with the protocols' implementations are open-source and freely available on-line for researchers to use.

We believe that the evaluation of the DTN protocol should continue given traces from other real-world applications. We are currently working towards providing a tool for acquiring such traces using OpenBeacon tags [49] as test-beds. Such a tool will help reduce the cost of collecting such traces and curating them into datasets that can be used by the research community for benchmarking DTN protocols and their applicability in real-world application scenarios. Moreover, it would be valuable to the research and engineering community to extend the reference benchmarking scenario to include other performance metrics, such as the number of initiated transmissions, average hop count and the number of dropped packets. This would allow one to further understand the behavior of the routing protocols under common application scenarios.

Acknowledgments: This work has been partially supported by the ICT Programme of the European Union under contract number ICT-2008-215270 (FRONTS) and the research project "Designing Human-Agent Collectives for Sustainable Future Societies" (C26A15TXCF) of Sapienza University of Rome.

Author Contributions: Khalil Massri, Andrea Vitaletti, Alessandro Vernata, and Ioannis Chatzigiannakis worked on the establishment of the reference architecture and the design of the benchmarking scenario. Khalil Massri, Andrea Vitaletti and Alessandro Vernata contributed on the overview of the relevant state of the art in comparison to the proposed reference architecture. Khalil Massri and Alessandro Vernata worked on the implementation of the DTN protocols and the extensions to the ONE simulator. Khalil Massri and Ioannis Chatzigiannakis contributed to the technical recommendations.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fall, K. A delay-tolerant network architecture for challenged internets. In Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Karlsruhe, Germany, 25–29 August 2003; pp. 27–34.
2. *Delay Tolerant Networking Research Group*; Internet Research Task Force, Internet Society: Reston, VA, USA, 2002.
3. Cerf, V.; Burleigh, S.; Hooke, A.; Torgerson, L.; Durst, R.; Scott, K.; Fall, K.; Weiss, H. *Delay-Tolerant Networking Architecture*; Heise Zeitschriften Verlag: Hannover, Germany, 2007.
4. Scott, K.; Burleigh, B. *Bundle Protocol Specification*; Internet Research Task Force, Internet Society: Reston, VA, USA, 2007.
5. Chatzigiannakis, I. Communication in Ad Hoc Mobile Networks Using Random Walks. In *Encyclopedia of Algorithms*; Springer: New York, NY, USA, 2008.
6. Wang, Y.; Wu, H. Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN): A New Paradigm for Pervasive Information Gathering. *IEEE Trans. Mob. Comput.* **2007**, *6*, 1021–1034.
7. Hui, P.; Crowcroft, J.; Yoneki, E. Bubble rap: Social-based forwarding in delay tolerant networks. In Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hong Kong, China, 26–30 May 2008; pp. 241–250.
8. Daly, E.M.; Haahr, M. Social network analysis for routing in disconnected delay-tolerant MANETs. In Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Montreal, QC, Canada, 9–14 September 2007; pp. 32–40.

9. Xian, Y.; Huang, C.T.; Cobb, J. Look-Ahead Routing and Message Scheduling in Delay-Tolerant Networks. *Comput. Commun.* **2011**, *34*, 2184–2194.
10. Burleigh, S.; Hooke, A.; Torgerson, L.; Fall, K.; Cerf, V.; Durst, B.; Scott, K.; Weiss, H. Delay-tolerant networking: An approach to interplanetary Internet. *IEEE Commun. Mag.* **2003**, *41*, 128–136.
11. Chatzigiannakis, I.; Mylonas, G.; Vitaletti, A. Urban pervasive applications: Challenges, scenarios and case studies. *Comput. Sci. Rev.* **2011**, *5*, 103–118.
12. Spyropoulos, T.; Rais, R.N.; Turletti, T.; Obraczka, K.; Vasilakos, A. Routing for disruption tolerant networks: Taxonomy and design. *Wirel. Netw.* **2010**, pp. 2349 – 2370.
13. Lenas, S.A.; Dimitriou, S.; Tsapeli, F.; Tsaoussidis, V. Queue-management architecture for delay tolerant networking. In *Wired/Wireless Internet Communications*; Springer: Berlin, Geramny, 2011; pp. 470–482.
14. Kate, A.; Zaverucha, G.M.; Hengartner, U. Anonymity and security in delay tolerant networks. In Proceedings of the Third International Conference on Security and Privacy in Communication Networks and the Workshops (SecureComm 2007), Nice, France, 17–21 September 2007; pp. 504–513.
15. Burgess, J.; Bissias, G.D.; Corner, M.D.; Levine, B.N. Surviving attacks on disruption-tolerant networks without authentication. In Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Montreal, QC, Canada, 9–14 September 2007; pp. 61–70.
16. Rhee, I.; Shin, M.; Hong, S.; Lee, K.; Kim, S.J.; Chong, S. On the levy-walk nature of human mobility. *IEEE/ACM Trans. Netw.* **2011**, *19*, 630–643.
17. A Community Resource for Archiving Wireless Data. Available online: http://www.crawdad.org/ (accessed on 25 March 2016).
18. Balasubramanian, A.; Levine, B.; Venkataramani, A. DTN routing as a resource allocation problem. *SIGCOMM Comput. Commun. Rev.* **2007**, *37*, 373–384.
19. Ali, S.; Qadir, J.; Baig, A. Routing protocols in Delay Tolerant Networks—A survey. In Proceedings of the 6th International Conference on Emerging Technologies (ICET), Islamabad, Pakistan, 18–19 October 2010; pp. 70 –75.
20. Keränen, A.; Ott, J.; Kärkkäinen, T. The ONE Simulator for DTN Protocol Evaluation. In Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Rome, Italy, 2–6 March 2009.
21. Liu, M.; Yang, Y.; Qin, Z. A Survey of Routing Protocols and Simulations in Delay-Tolerant Networks. In *Wireless Algorithms, Systems, and Applications*; Springer: Berlin, Geramny, 2011; pp. 243–253.
22. Dang, H.; Wu, H. Clustering and cluster-based routing protocol for delay-tolerant mobile networks. *IEEE Trans. Wirel. Commun.* **2010**, *9*, 1874–1881.
23. Scott, J.; Gass, R.; Crowcroft, J.; Hui, P.; Diot, C.; Chaintreau, A. CRAWDAD dataset cambridge/haggle (v. 2009-05-29), doi:10.15783/C70011, May 2009. Available online: http://crawdad.org/cambridge/haggle/20090529 (accessed on 25 March 2016).
24. Amici, R.; Bonola, M.; Bracciale, L.; Rabuffi, A.; Loreti, P.; Bianchi, G. Performance Assessment of an Epidemic Protocol in VANET Using Real Traces. *Procedia Comput. Sci.* **2014**, *40*, 92–99.
25. Mathurapoj, A.; Pornavalai, C.; Chakraborty, G. Fuzzy-spray: Efficient routing in delay tolerant ad-hoc network based on fuzzy decision mechanism. In Proceedings of the 18th International Conference on Fuzzy Systems, Jeju Island, Korea, 20–24 August 2009; pp. 104–109.
26. Pasztor, B.; Musolesi, M.; Mascolo, C. Opportunistic Mobile Sensor Data Collection with SCAR. In Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, Pisa, Italy, 8–11 October 2007; pp. 1–12.
27. Massri, K.; Vitaletti, A.; Vernata, A. A complete study and evolution about DTN routing protocols and environments. Available online: http://code.google.com/p/evolving-dtn-routing/ (accessed on 25 March 2016).
28. Dias, J.A.; Isento, J.N.; Soares, V.N.G.J.; Farahmand, F.; Rodrigues, J.J.P.C. Testbed-based Performance Evaluation of Routing Protocols for Vehicular Delay-Tolerant Networks. In Proceedings of the IEEE GLOBECOM Workshops (GC Wkshps), Houston, TX, USA, 5–9 December 2011; pp. 51–55.
29. Burgess, J.; Gallagher, B.; Jensen, D.; Levine, B.N. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In Proceedings of the 25th IEEE International Conference on Computer Communications INFOCOM, Barcelona, Spain, 23–29 April 2006.

30. Lindgren, A.; Phanse, K. Evaluation of Queueing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks. In Proceedings of the First International Conference on Communication System Software and Middleware, New Delhi, India, 8–12 January 2006; pp. 1–10.
31. Spyropoulos, T.; Psounis, K.; Raghavendra, C.S. Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility. In Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops, White Plains, NY, USA, 19–23 March 2007; pp. 79–85.
32. Soares, V.N.; Rodrigues, J.J.; Farahmand, F. GeoSpray: A geographic routing protocol for vehicular delay-tolerant networks. *Inf. Fusion* **2014**, *15*, 102–113.
33. Nelson, S.C.; Bakht, M.; Kravets, R.; Harris, A.F., III. Encounter: Based Routing in DTNs. *SIGMOBILE Mob. Comput. Commun. Rev.* **2009**, *13*, 56–59.
34. Leontiadis, I.; Mascolo, C. GeOpps: Geographical Opportunistic Routing for Vehicular Networks. In Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007), Helsinki, Finland, 18–21 June 2007; pp. 1–6.
35. Vahdat, A.; Becker, D. Epidemic Routing for Partially-Connected Ad Hoc Networks. Master's Thesis, Duke University, Durham, NC, USA, 2000.
36. Spyropoulos, T.; Psounis, K.; Raghavendra, C.S. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking, Philadelphia, PA, USA, 26 August 2005; pp. 252–259.
37. Lindgren, A.; Doria, A.; Schelén, O. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2003**, *7*, 19–20.
38. Dubois-Ferriere, H.; Grossglauser, M.; Vetterli, M. Age matters: Efficient route discovery in mobile ad hoc networks using encounter ages. In Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing, Annapolis, MD, USA, 1–3 June 2003; pp. 257–266.
39. Tan, K.; Zhang, Q.; Zhu, W. Shortest path routing in partially connected ad hoc networks. *IEEE Glob. Telecommun. Conf.* **2003**, *2*, 1038–1042.
40. De Oliveira, E.C.R.; de Albuquerque, C.V.N. NECTAR: A DTN routing protocol based on neighborhood contact history. In Proceedings of the ACM Symposium on Applied Computing, Honolulu, HI, USA, 9–12 March 2009; pp. 40–46.
41. Sandulescu, G.; Nadjm-Tehrani, S. Opportunistic DTN routing with window-aware adaptive replication. In Proceedings of the 4th Asian Conference on Internet Engineering, Bangkok, Thailand, 18–20 November 2008; pp. 103–112.
42. Boldrini, C.; Conti, M.; Jacopini, J.; Passarella, A. HiBOp: A History Based Routing Protocol for Opportunistic Networks. In Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks, Espoo, Finland, 18–21 June 2007; pp. 1–12.
43. Mangrulkar, R.; Atique, M. Routing protocol for Delay Tolerant Network: A survey and comparison. In Proceedings of the IEEE International Conference on Communication Control and Computing Technologies (ICCCCT), Ramanathapuram, India, 7–9 October 2010; pp. 210–215.
44. Shah, H.; Kosta, Y.P. Evolution of Various Controlled Replication Routing Schemes for Opportunistic Networks. In *Advanced Computing*; Springer: Berlin, Germany, 2011; pp. 338–347.
45. Chatzigiannakis, I.; Nikoletseas, S.; Spirakis, P. On the Average and Worst-Case Efficiency of Some New Distributed Communication and Control Algorithms for Ad-Hoc Mobile Networks. In Proceedings of the 1st ACM International Annual Workshop on Principles of Mobile Computing, Newport, RI, USA, 29–30 August 2001; pp. 1–19.
46. Chatzigiannakis, I.; Nikoletseas, S.; Spirakis, P. Distributed communication algorithms for ad hoc mobile networks. *J. Parallel Distrib. Comput.* **2003**, *63*, 58–74.
47. Lee, K.; Hong, S.; Kim, S.J.; Rhee, I.; Chong, S. SLAW: A New Mobility Model for Human Walks. In Proceedings of the 29th IEEE International Conference on Computer Communications INFOCOM, Rio de Janeiro, Brazal, 19–25 April 2009; pp. 855–863.
48. Da Cunha, F.D.; Alvarenga, D.A.; Viana, A.C.; Mini, R.A.F.; Loureiro, A.A.F. Understanding Interactions in Vehicular Networks Through Taxi Mobility. In Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks, Cancun, Mexico, 2–6 November 2015; pp. 17–24.

49. OpenBeacon Active RFID Project; Available online: http://www.openbeacon.org/ (accessed on 25 March 2016).