Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2011, Article ID 892038, 14 pages
doi:10.1155/2011/892038

Research Article

Opportunistic P2P Communications in Delay-Tolerant Rural Scenarios

Marcel C. Castro,[1] Laura Galluccio,[2] Andreas Kassler,[1] and Corrado Rametta[2]

[1] *Computer Science Department—Telematics, Karlstad University, Sweden*
[2] *Dipartimento di Ingegneria Informatica e delle Telecomunicazioni, universitá di Catania, Italy*

Correspondence should be addressed to Laura Galluccio, lauragalluccio@gmail.com

Received 16 May 2010; Revised 13 September 2010; Accepted 14 October 2010

Academic Editor: Andrew T. Campbell

Opportunistic networking represents a promising paradigm for support of communications, specifically in infrastructureless scenarios such as remote areas communications. In principle in opportunistic environments, we would like to make available all the applications thought for traditional wired and wireless networks like file-sharing and content distribution. In this paper, we present a delay-tolerant scenario for file sharing applications in rural areas, where an opportunistic approach is exploited. In order to support communications, we compare two peer-to-peer (P2P) schemes initially conceived for wireless networks and prove their applicability and usefulness to a DTN scenario, where replication of resources can be used to improve the lookup performance and the network can be occasionally connected by means of a data mule. Simulation results show the suitability of the schemes and allow to derive interesting design guidelines on the convenience and applicability of such approaches.

1. Introduction

Opportunistic networking has attracted the interest of researchers in the last years. The use of this paradigm becomes critical in challenging scenarios like satellite applications and rural communications in emerging countries like India or Africa, where the lack of an infrastructure makes communications almost impossible.

Delay-tolerant (DTN) communications are thus the natural choice for a networking paradigm where nodes can be disconnected from the Internet for the majority of the time and exchange of data can take very long time. DTN communications have been usually considered in the perspective of supporting data delivery, for example in sensor applications, where data mules are introduced with the purpose of collecting the data monitored by remote devices and delivering them to a collection center [1].

In emerging countries numerous projects aimed at rural poverty alleviation have been proposed. For example the Sustainable Access in Rural India (SARI) program [2], inaugurated in 2001, consists of disseminating more than 80 rural Internet kiosks distributed in the Madurai area of Tamil Nadu in India. However, not all villages can be served by these kiosks and thus, in parallel, exploiting an opportunistic approach, the Computers on Wheels (COW) project [3] has been carried out in India as well since 2003. In this case, a set of motorcycles equipped with an Internet-connected laptop travel between very remote villages to collect requests for Internet access and support users' communications during the limited time the motorcycle stops at the village. Similar initiatives have been recently carried out also in Africa [4, 5] where motorcycles have been substituted by buses or cars. For example, solar powered kiosks will be deployed in the Serengeti area, and equipped with an Internet connection point. Here, people can attach their handheld devices for recharge and access the network while in their proximity. In this case, the data mule approach is inverted since remote disconnected nodes are mobile while kiosks, that is, data mules, are static. Other scenarios where kiosks are static is the Air Jaldi project [6], where more than 30 mesh routers around Dharamsala in rural India have been employed to provide connectivity to mobile clients when in range of the mesh router. Concerning satellite communications, the opportunistic networking paradigm is used for deep space communications [7].

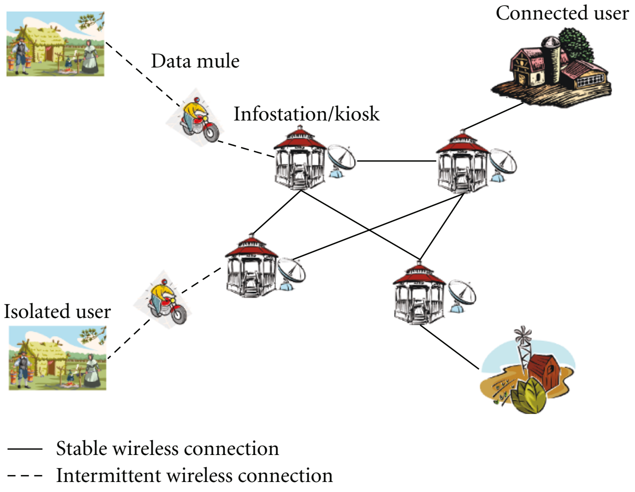

Figure 1: Opportunistic networking scenario with static Infostations and data mules.

With such a communication scenario in mind, in this paper we consider a delay-tolerant scenario where users move freely within a disconnected rural area. We assume a static infostation deployment is available, for example, using wirelessly connected Internet kiosks, allowing users to connect to the Internet while being located in their closest proximity. Users far from the infostations cannot connect to the Internet unless a data mule comes close to them and a multihop communication can be set up towards the closest infostation. We assume that infostations are connected with each other using some form of wireless network. For this purpose, for example, mesh networks can be used, which are becoming nowadays common in rural areas. Figure 1 shows this reference scenario. In such an environment, rural communications can be allowed during the limited time the isolated user comes into proximity of a data mule which can both perform as a simple relay towards the connected backhaul, or store the requests on the isolated node's behalf and process them while moving.

Resource requests performed by remote users can be:

(i) issued and retrieved at any time while the user is close to the infostation. In this case, resource search and retrieval are not significantly constrained,

(ii) issued and retrieved by an isolated remote node during the limited time the data mule comes into its proximity when a multihop communication can be set up towards the infostation. This could lead to two different situations. In the first case, when the resource search and retrieval is fast, the resource can be searched for and downloaded during the limited proximity time. In the second case, if the search is not fast enough, the resource will be retrieved next time the data mule comes back. In this case, a pure delay-tolerant paradigm is employed and only reliability constraints are met.

In order to locate resources distributed within the network, various schemes have been proposed in the context of peer-to-peer applications, also considering wireless and multihop networks. For example, Pastry [8], Bamboo [9], Viceroy [10], Georoy [11], Chord [12], and so forth, are only some of the most common approaches proposed. However, in delay-tolerant application scenarios, opportunistic intercontact intervals between mobile and remote users should be exploited at maximum since they represent communication chances. To improve the performance of the network, as proposed already by previous literature in the field, resources available can be replicated so that multiple copies of the same file are distributed by exploiting the mobile users' movements and the opportunistic intercontacts with the infostations or kiosks.

In this paper, we present a performance evaluation and comparison study of two P2P resource management approaches in the opportunistic scenario described before. We identify a tradeoff between search-retrieval efficiency and algorithm complexity. The impact of using these P2P approaches in such scenario is estimated through ns2 simulations. The main contributions of this work are related to testing of the performance of two efficient P2P approaches conceived for wireless networks and appropriately extended to cope with the constraints of a DTN scenario. Also replication and opportunistic networking were addressed and appropriate protocol elements developed for Georoy which is one of the two protocols being analyzed.

The rest of this paper is organized as follows. In Section 2 we discuss related literature in the field of opportunistic/delay-tolerant networking and P2P networks. In Section 3 we present in detail the addressed scenario. Section 4 gives an overview of two P2P algorithms which we will later evaluate in more detail: Bamboo and Georoy. In Section 5 we introduce the replication mechanism which can be exploited for improving the efficiency of the search procedure. In Section 6 we discuss the suitability of the discussed algorithms to a DTN scenario. In Section 7 we compare the performance of the two protocols and derive some insights on their behavior. Finally, in Section 8 some conclusions are drawn and a discussion about future work is presented.

2. Related Work

In this section we discuss recent literature in the field of opportunistic and delay-tolerant networking and P2P algorithms.

2.1. Opportunistic and Delay-Tolerant Networking. An opportunistic network is a type of challenged network where intermittent network contacts are met and link performance are variable and unstable. In general in these networks stable end-to-end paths do not exist since nodes can be isolated most of the time and paths may frequently break up. To cope with these problems while supporting communications, a storecarry-forward approach can be used where intermediate nodes keep the message while the connectivity is down. This requires that applications are delay-tolerant. Moreover, the use of an opportunistic paradigm allows to foresee a process of resource propagation during occasional intercontacts between nodes.

ZebraNet [13] is an example of DTN networking, which tracks animal movements across a wide area. Collars carried by animals work like peer-to-peer devices which communicate to deliver logged data to monitoring centers. DTN networking is also dealt with in an analytical perspective in the Pocket Switched networks [14] where intercontact times among pairs of nodes are analyzed in real human mobility scenarios. Similar studies aimed at characterization of social interactions have been also carried out at MIT in the context of the reality mining project [15]. Also the Haggle project [16] proposed a networking architecture along with a set of protocols and description languages to enable communication in intermittent network connectivity scenarios.

Concerning the network layer, two routing approaches are common in opportunistic scenarios: forwarding and flooding. In forwarding, intermediate nodes relay a single copy of the packet over several hops towards the final destination. The difference among the various forwarding approaches relies in the methodology used for selecting the best path for forwarding data: direct-transmission [17, 18], location-based transmission [19, 20] or using an estimation-based approach [21, 22]. The forwarding approach has typically low overhead in terms of packets circulating in the network but can suffer for low packet delivery ratio and long delivery delays. On the contrary, the flooding based schemes are more robust but can add significant overhead into the network by having multiple copies of a packet traversing the network.

In opportunistic networks, a connection-oriented transport layer protocol such as TCP requires reengineering due to frequent disruptions and intermittent end-to-end connectivity. For example, the Licklider Transmission Protocol (LTP) and its evolutions have been introduced in order to cope with retransmissions in high latency environments such as the challenged ones. Typically, a new protocol layer is required to be identified and located in between the application and transport layers. This protocol, denoted as bundle layer [23, 24], allows each node to act as both a router or a gateway to transfer messages across different regions. In this way the problem of supporting traditional applications where the end-to-end source-destination connections do not exist can be overcome. At the Bundle layer, functionalities of storing-carrying-forwarding are considered and employed for multicast and anycast [25–27]. Finally, concerning the application layer, support for traditional applications such as Web and email is not possible since the underlying transport protocols do not work properly in challenged opportunistic environments. As a consequence, in [28] the use of SMTP proxies is introduced to hide disruptions among users. Emails are thus sent in bundles into the opportunistic network and carried to a mail gateway which forwards and receives the mail between the infrastructured and the opportunistic networks. In [29], an Internet proxy is used to collect search engines and prefetch web pages. The user query is stored until the mobile node will contact the proxy after a disruption.

2.2. P2P Algorithms. P2P communication protocols have been primarily designed to work in wired scenarios. Napster [30] was the first approach proposed for P2P applications although it was not purely P2P since it exploited a centralized set of servers for resource indexing. In spite of its evident limitations, Napster paved the way to other schemes like Gnutella [31] in its various versions that did employ a real P2P philosophy using a virtual overlay flooded for resource searches. However, use of flooding caused scalability problems. Accordingly, more flexible solutions were invented. As an example Kazaa [32] used a hybrid approach considering that peer nodes are separated into Super Peers (SP) and Leaf Peers (LP). While SPs publish resources in a distributed catalog, LPs provide the resources. In Kazaa, SPs represent an unstructured overlay where resources can be located by flooding requests into the network. While the use of an organized overlay causes a higher flexibility in the network, flooding is costly in terms of scalability. As a consequence, Distributed Hash Table-(DHT-) based solutions have been proposed. DHTs offer an indexing service by mapping each resource and each node storing the resource on a certain key assigned through a specific methodology. The one-way hash function leads to every SP node being responsible for a range of keys and having a virtual link with a subset of network nodes. When someone requests a key to a node, a node compares its own ID with the key and, if it falls in its node range, it replies to the requester, otherwise the request is forwarded to the neighbor whose ID is the closest with respect to the searched key. Chord [12], Pastry [8], Tapestry [33], and Viceroy [10] are all solutions that exploit a DHT approach. They differ in the way they build and maintain the structure of the logical overlay. For example, Chord uses a logical ring where every node has an assigned ID and is responsible for all the keys between its ID and its predecessor ID (which is known as well as the successor ID). Moreover, in order to speed up the search process, a *Finger Table* is used to connect the node to other nodes in the network.

After Chord, other robust algorithms were proposed like *Pastry* [8] and *Tapestry* [33]. These protocols follow basically the same methodology for the next-hop choice, that is, the node with the longest common prefix with respect to the searched key is selected, but exploit different routing mechanisms in the overlay.

Common features of these schemes are that the size of the routing tables typically increases logarithmically with the size of the network. In order to provide an upper bound to the lookup search performance, in Viceroy [10] a combination of a unit ring topology and a butterfly network [34] topology is proposed. In such a way, a lookup performance of $O(\log n)$ can be achieved with a routing table which contains at most 7 entries. However, all the above-mentioned solutions employ a logical overlay which is completely independent of the existing physical network and, thus, in general, even if two nodes are physically close, they can be far away in the overlay. This leads to a problem when such overlays are deployed over resource scarce wireless networks such as multihop wireless ad hoc or mesh networks. Here, it is crucial to minimize the number of physical hops as this directly impacts the achievable delay and packet loss. Other problems which can arise are related to high churn rates when there are SP nodes who frequently attach and detach.

Several enhancements have been proposed to DHTs in order to improve performance over resource scarce networks. Probably the most prominent approaches in this category are Georoy [11] and Bamboo [9].

In the rest of this paper we will compare the performance achieved by these schemes. Accordingly, in the following sections we will describe these two algorithms more in detail.

3. Scenario

In this paper we address an opportunistic scenario where resources are disseminated across the network and nodes can access them. In the illustration of the scenario, we refer to what is shown in Figure 1 where infostations are deployed statically, which allow to set up the resource search. Infostations are connected typically using some wireless links and connections among infostations are considered stable. As an example, a mesh network can provide backhaul connectivity between the Infostations, where every mesh router has the functionality to setup the resource search. Also, there is a certain number of peripheral nodes which can provide and/or search for resources. Some of these peripheral nodes can be isolated and not in the range of any infostation so that their resources cannot be shared and their requests cannot be served directly. We assume that one or more data mules can move around and serve the isolated nodes once they come in their closest proximity. Obviously, the mobile node remains in the proximity of the isolated node for a limited time interval during which resource search must be performed and the resource should be provided to the requesting node. If these two processes are not successfully completed during the limited proximity time, the isolated node cannot exchange data with the rest of the network. A solution to this problem could exploit a delay-tolerant paradigm. In fact, the mobile node can cache the lookup request as issued from the isolated node and keep on performing the lookup during its tour throughout the network. Once the lookup request is answered successfully, the data mule retrieves the resource and stores it until it comes again in proximity of the isolated node which can then be served.

In order to implement the above-presented scenario, we have chosen to compare the performance of two P2P protocols for wireless networks, appropriately extended to cope with the opportunistic networking scenario. More in depth, Bamboo and Georoy, the two protocols considered that will be described in the following sections, retrieve resources according to a distributed mechanism where, to speed up the lookup process and make it suitable to an unreliable scenario like the one addressed by our study, a replication methodology for managing multiple copies of the same resource has been introduced. Speeding up the lookup process is important as the data mule is in close proximity of a given infostation only for a limited time period. This time interval during which the lookup request/response needs to be completed depends on the speed of the data mule and the mobility pattern.

4. Opportunism and P2P Systems

In this section we will preliminarily describe the two considered P2P protocols. Then, in the next section, we will discuss the replication mechanisms used to increase the chances of having a successful resource lookup, of the two algorithms in opportunistic scenarios.

4.1. Bamboo

4.1. Bamboo. Bamboo [9] is inspired by previous DHT schemes such as Pastry [8] and aimed at reducing congestion due to large management traffic. While Bamboo is based on the routing logic of Pastry, management of overlay structure is different in the aim of being more scalable in dynamic environments.

To maintain the network structure, Bamboo uses two sets of neighbor information at each node: *leafset* and *routing table*. The leafset consists of successors and predecessors that are numerically closest in the key space. While two nodes may be neighbors (in the leafset) in the overlay, they may be physically far away. When performing a query, the latter is forwarded until a node which has the key in its leafset to ensure correct lookup is reached. To improve lookup performance, a routing table is used, which is populated with nodes that share a common prefix. Accordingly, routing table lookups are ordinary longest prefix matches. The routing table is of size $\log_{2^b} N \times (2^b - 1)$, where N is the number of nodes in the network and b is a configuration parameter (e.g., $b = 2$).

When data is stored in the system using the *put* command, the data is routed using the DHT to the node primarily responsible for storing the data. The major difference between Pastry and Bamboo is the way they handle management traffic. In Pastry, management is initiated when a network change is detected, while in Bamboo management traffic is periodic, regardless of network status. While reactions to changes in the routing layer operate on very small timescale, reactions to changes in overlay structure are not so fast. However, the approach to use periodic updates has shown to be beneficial during churn [9], since it does not cause management traffic bursts during congestion. Such traffic bursts can increase packet loss probability, lead to management messages being dropped and cause other overlay network problems.

In standard configurations, Bamboo optimizes latency. It is important to note that an optimized routing table does not influence lookup correctness, but only lookup latency [35]. As wireless networks are rather limited in bandwidth, a balance between overlay lookup efficiency and management traffic overhead is important [36].

4.2. Georoy. The Georoy algorithm [11] is a location-aware variant of the Viceroy algorithm [10] briefly described in Section 2. The main target of Georoy is to build an overlay network that can provide accurate and efficient resource lookup in an ad hoc wireless network, supporting either node mobility and resource adding or removing. Using a geographic aware hash function, Georoy is able to obtain a very small *stretch factor*, that is, the ratio between the hop distance of the path traversed by the query in order to find the node and the number of hops traversed in the physical network from the searching node to the searched one. The stretch factor gives a measure of the discrepancy between the physical hops traversed during resource lookup and those that would have been traveled going directly to the final destination using minimum hop count routing.

As a main difference with Chord and others, Georoy does not use a flat node topology, but employs a two level hierarchy with two different kinds of nodes: *Leaf Peers (LP)* which share and request resources by querying their associated super peers and *Super Peers (SP)* which provide the distributed resource catalog and are used by LPs to publish and request resources.

Typically, SPs are wireless routers which are placed in the network and do not move; LPs are mobile nodes that can move and stay connected via a handoff mechanism like in cellular networks. In Georoy, the DHT is managed only by SPs which are also responsible for the overlay construction and maintenance; so the IDs in the DHT are assigned only to these nodes. When a LP wants to share a resource it must associate this resource with a key provided by its SP according to a distributed hash function. Resource IDs are mapped in the same ID space of SPs, that is, $[0,1]$ so, both the SPs ID space and the resource keys space are mapped in the same interval $[0,1]$. Each resource key is managed by the SP with the smallest ID larger than the key ID so that each SP is responsible of all IDs between its own one and the one of its predecessor (which is known).

In order to provide geographic awareness, a mapping function is proposed which gives a SP an ID depending on its physical x and y coordinates. To explain this function we assume that nodes are deployed in a square region of side s, so all SPs are located in $R = [0,s)X[0,s)$. The mapping function $\mathcal{M}$ is defined as follows:

$$\mathcal{M}(x,y) = \begin{cases} \dfrac{x\Delta}{s^2} + \left\lfloor \dfrac{y}{\Delta} \right\rfloor \dfrac{\Delta}{s} & \text{if } \left\lfloor \dfrac{y}{\Delta} \right\rfloor \text{ is even,} \\[2ex] \dfrac{(s-x)\Delta}{s^2} + \left\lfloor \dfrac{y}{\Delta} \right\rfloor \dfrac{\Delta}{s} & \text{if } \left\lfloor \dfrac{y}{\Delta} \right\rfloor \text{ is odd,} \end{cases} \tag{1}$$

with $0 < \Delta < s$.

When a node joins the network, it first computes its ID using the function described. Then it chooses a level at random and joins the ring through lookup predecessor and successor. Finally, after establishing unit and level rings, butterfly connections are set up. (For more details on Georoy procedures please refer to [11].)

5. Resource Replication

In P2P networks, the lookup procedure can take very long time when the size of the network increases. This is especially the case when deployed over multihop wireless networks as for each physical hop, the lookup message needs to contend again for the medium. Therefore, reducing the total number of physical hops traveled directly impacts on the achievable performance. Also, when only a single copy of the resource is available in the network. (For worth of simplicity, in the following we will assume that a LP node provides only a single resource. Generalization to the case of multiple resources provided by a node is straightforward.) The provider node could become congested if multiple peers request the resource. Moreover, if the responsible node crashes, the resource will be no longer available. Accordingly, replication of resources can be beneficial since it allows to balance the network traffic among the different replicas' providers. This can reduce the delivery delay both in case of resource lookup and resource delivery. In fact, when more copies of a resource are available in the network, it is expected that the resource can be located in the closer proximity of the requesting node. While a replication mechanism for Bamboo has already been specified, in this paper, we develop a replication strategy for Georoy which we will describe in the following.

5.1. Resource Replication in Georoy. In Georoy when a LP storing a resource and located closer to an infostation node, denoted as SP, moves it can decide to replicate its resources with a given probability, P_R, at its old SP. For example, if the LP denoted as D, previously located closer to the SP denoted as B moves, it can decide if leaving a copy of its resource in B's area or not according to a given probability P_R. Then, when D moves and comes into proximity of a new SP called C, its resource becomes again available. So the number of copies of each D's resource into the network are given by $1 + N_{\mathrm{SP}v} \cdot P_R$ where $N_{\mathrm{SP}v}$ represents the number of different SP nodes visited during D's tour in the interest area. In fact, if a node visits many time the same SP, it does not try to replicate its resource at the same node everytime but just once. Replication of a resource requires an update at the Home SP managing the range of keys to which the resource belongs. When the LP node D moves and goes out of the coverage area of its responsible SP, if replication happened, B will ask one of the other LPs in its coverage area to store the copy of the resource. This will be done through a `put_resource` message. Then B will contact through a lookup the corresponding Home SP storing the range of keys the resource belongs to and notifies the availability of a replica of that resource at its site. When the node D comes into the proximity of another SP C, it will notify its catalog of resources and the SP C will keep the Home SP updated through a lookup operation.

When a lookup for that resource will be generated, it will be forwarded throughout the Georoy overlay as usual. Two cases can happen.

(i) *Case 1*: the resource is available at one of the SPs traversed along the path going to the Home SP responsible for that resource. In this case, the lookup is positively answered before reaching the Home SP and the resource is located fastly.

(ii) *Case 2*: the lookup is forwarded till the Home SP is met but the resource is not located before reaching the Home SP. In this case, the Home SP owns a list of the SP nodes that have the resource in their catalog. Accordingly, based on the ID of the node who issued the lookup, the Home SP answers with the ID of the SP among those which store the resource that is closer to the ID of the requester. This is because closer IDs in the logical space mean also closer physical location due to the intrinsic property of the Georoy mapping.

Observe that replication implies an increase in the rate of availability of a resource in the network but could cause an increase also in the overhead at network nodes. Accordingly, a mechanism to control the number of replicas of a given resource available in the network should be considered. To this purpose, in Georoy we assume that the oldest copies of a resource are deleted after a time out so that a maximum number of replicas for a resource R_M can be found into the network. To implement this control, Georoy has been modified in the following way. When the Home SP of a resource, which is aware of the number of copies of a resource available in the network and the time they were generated, sees that R_M copies are currently available into the network, as soon as it receives another notification for a new copy of the resource, will accept it and contact the responsible SP for the oldest copy to ask for deleting the resource from the catalog. To this purpose a `delete_resource` message will be sent. The Responsible SP, upon receiving such a notification, contacts the LP storing the copy and, if it is still in its coverage area, asks for deleting the resource. Accordingly, a `delete_replica` message will be exploited. If the LP moved, the resource is considered no longer available in any case.

To be sure that the available replicas of the resource are still valid, each responsible SP periodically interrogates the LP that is supposed to store the copy of the resource using a beaconing-like approach. If the LP moved without notification, the status of the copy is updated as *parked* at the Home SP and managed as specified in the following section. Accordingly the number of copies of a resource in the network is kept updated.

5.2. Resource Replication in Bamboo. In Bamboo, a replication mechanism is already incorporated. This is quite simple with respect to Georoy and provides incremental scalability. Basically, a node holding a given resource also caches it within some of its leafset neighbors. This is done according to a number of desired replicas. To this purpose, `put` messages are generated by the node to selected peers among its successors and predecessors. For example, if the desired number of replicas is set to 4, the node generates 4 Bamboo `put` messages destined to 2 random successors and 2 predecessors, achieving a total of 5 resource copies in the network. Therefore, the amount of overhead in the network increases with the number of replicas. It is also important to note that the maximum amount of replicas is given by the total number of nodes in the leafset (i.e., number of successors and predecessors). This means that in the default scenario where the number of leafsets is configured to 7, a maximum of 15 copies of the resource will be available in the network.

When an existing node leaves the system, it takes the data it has stored with it. Therefore, the redundancy given by the replication strategy guarantees that the resource will be still available in the remaining leafset neighbors. In order to keep the distributed storage consistent, data storage updates are also applied by Bamboo, where a node periodically picks a random node in its leafset and synchronizes the stored data with it. The correspondent node calculates the set among its stored data that should be stored at the peer node, sending this data to it, including the hash values of the data.

For certain applications, the number of desired replicas can cause large demands for storage space. This can turn into serious scalability problems when disseminating these replicas to many nodes in the leafset.

6. Delay-Tolerant Networking (DTN) Paradigm in P2P Schemes

To support P2P networking in opportunistic or DTN scenarios, the following situations should be addressed:

(i) the node who invoked the lookup moved or is no longer connected to the network and the lookup procedure should be still completed,

(ii) the node who owns a resource is no longer accessible but the resource should be still available for download.

These aspects are explicitly addressed by the Georoy protocol and the modifications introduced in the previous section are detailed in the following.

6.1. LP Joining/Leaving in Georoy. Once a SP node B is connected, it can accept LPs connections and can route lookup requests. A LP D, upon entering the network, needs to invoke a join procedure to register its available catalog of resources. Accordingly, listening on the wireless interface, D selects the SP with the best received quality which becomes its responsible SP, and registers by providing it with the list of the resources it is willing to share. Such information is maintained up-to-date by B in a local database of available resources. Also, for each LP resource, there is a Home SP which manages the pointer to the physical location of the resource, that is, the current responsible SP to which the leaf peer D is currently connected, and the Home SP does not change as the LP storing the resource moves throughout the network. Databases are managed in a distributed way in the sense that all SP nodes own a database listing LP nodes currently in their coverage area and the resources associated. In addition each SP stores also a list of the resources it is Home SP for and the associated list of nodes which store a copy of each resource together with a timestamp which says when the replica was generated.

Everytime the LP moves, the responsible SP must inform the related Home SP about its new location and its resources, both available and parked. When a LP D leaves the network, the list of resources available in the network has to be updated. Such update is necessary in order to maintain correct information of resources that are currently available in the network.

Before leaving the network, node D notifies its responsible SP, B, so that it puts the resource shared by node D into *park mode* through an appropriate tagging of the entry in its local database. Also, the Home SP must be informed that D is leaving the network so that it puts the resource hold by D into *park mode*.

As a consequence, if D will be again available within a short time, the resource will only be tagged as *available* at B and at the Home SP. In this way the signaling in the network is maintained at a minimum level.

Resources that are in *park mode* for a very long time interval are removed from the local resource database, and considered as no longer available.

To cope with conditions when, due to a failure, a LP node detaches without notification to its responsible SP, a beaconing procedure is activated. More in depth, the SP sends periodically an *OK-message* to the LP. If, after a time T_{up} the LP D does not answer, the resource of D is labeled as in *park mode* and the Home SP is informed. When a lookup for a resource labeled as in park mode is issued (i.e., the node storing it detached from the network), the following situations can be met:

(i) if the replication is used, the resource can be found at another node. The Home SP will thus manage transparently such a condition,

(ii) if no replication is used or other replicas are not currently available, the lookup will be delayed for a time interval set depending on application requirements. If the entry will not be updated at the Home SP (i.e., the resource is still in park mode), a denial will be issued as an answer to the lookup.

6.2. LP Handoff Management. Suppose that a certain LP D, which was formerly associated with a responsible SP B, migrates in the coverage area of another SP, C. In this case the following operations are required: (i) informing the Home SP that from now on the resource could be located (also) at node C, (ii) deleting the resources stored by D from the catalog of the resources locally available at B (if no replication has been performed), (iii) inserting the resources stored by D into the catalog of the resources locally available at C; and (iv) informing all the nodes that are currently downloading resources from D, if any, that this node has moved to a new position. To this purpose the Home SP contacts them using an `update_LP_position` message and notifying the ID of the SP in which coverage the resource can be found. Accordingly, a lookup to this node will be issued by interested nodes.

Observe that the use of the Home SP mechanism increases efficiency significantly when handoff occurs. In fact, besides local signaling between the leaf peer D and the past and current responsible SPs, B and C, only a location update must be sent to the Home SP. Instead, if the Home SP mechanism was not used, the location update should have been transferred to all SPs that contain the location information about node D.

7. Performance Results

In this section we compare the performance of the two protocols, Bamboo and Georoy in different conditions. We want to better understand their behavior by means of a comparison using two significant scenarios representing the static backhaul of wireless nodes: grid and random scenarios. Here, a number of SP nodes (i.e., Infostations) are placed within an area of a certain size. The Infostations are static and do not move during the simulations. We vary the number of such stations between 25 and 225. Ns2 v2.26 [37] simulations were run considering a transmission range of 200 m, a carrier sense range of 250 m, an area which size $\mathfrak{R}$ depends on the number of SP nodes as $\mathfrak{R} = N_{\text{SP}} \cdot 10^4\,\text{m}^2$ and a distance between two SPs in the grid topology equal to 100 m. Routing between the connected Infostations uses AODV-UU [38] but different choices are possible. In the random topology, nodes are thrown randomly in the area. We consider infinite buffer space on the replication nodes. We make such choice because if the buffer size is limited, achievable performance may largely depend on buffer replacement strategies, which is a problem outside the scope of this paper. In the random topology case, for each scenario identified by the number of nodes, we tested 5 different random topologies and for each topology we performed 100 random lookups. Average values and confidence intervals (when applicable) were reported for the following performance metrics being investigated:

(i) number of logical hops traveled in the overlay network to perform a lookup for a specific resource,

(ii) corresponding number of physical hops traveled in the physical network to perform a lookup for a specific resource as a consequence of the logical path followed,

(iii) lookup delay needed for the lookup to reach the node who stores information about the requested resource. We only consider here correctly completed lookups.

(iv) percentage of lookups correctly completed,

(v) stretch factor, that is, the ratio between the number of physical hops needed to complete the lookup as a consequence of the logical hops traversed and the number of physical hops going end-to-end according to a shortest path approach.

In the first part of the evaluation, we focus on the impact of the network size on the scalability of the lookup procedure. We then evaluate the impact of the replication technique. Finally, we evaluate the impact of the use of a data mule on the achievable performance in terms of resources download.

7.1. Impact of Network Size in Grid and Random Topologies. In Figure 2 we show the number of logical hops traveled when employing the two algorithms. By comparing the results we observe that, in general, Bamboo results in a smaller number of logical hops as compared to Georoy. This is related to the fact that the amount of overlay routing information used by Bamboo (i.e., leafset and routing table) is higher if compared to Georoy which limits the number of existing logical links to 7. Therefore, Bamboo can more easily identify a requested resource as it has more routing information available. In contrast, the number of physical hops mainly impacts on the lookup performance.

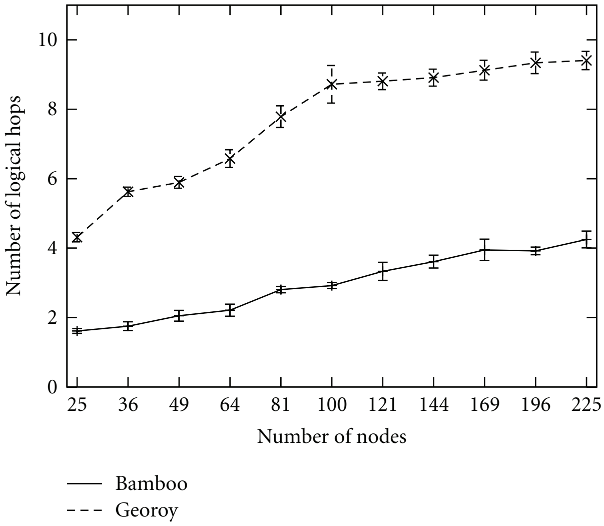

Figure 2: Comparison between the number of logical hops in Georoy and Bamboo in a grid topology.

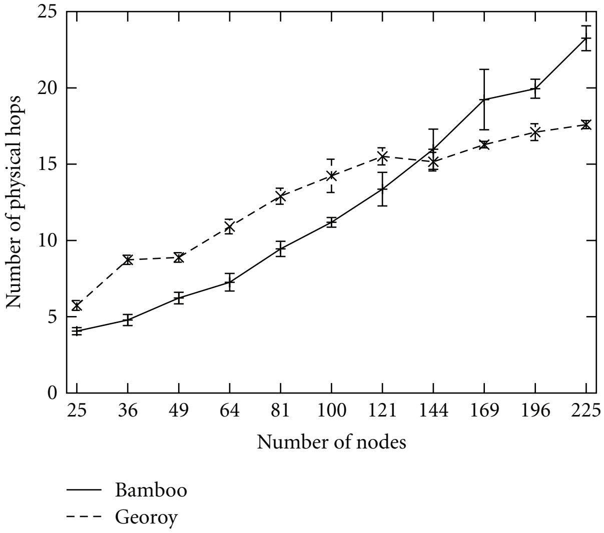

Figure 3: Comparison between the number of physical hops in Georoy and Bamboo in a grid topology.

This is because this parameter determines the number of forwarding operations a packet needs to undergo in the wireless multihop network to reach the destination (i.e., the node holding the resource). As the network size grows, also the number of physical hops needed to complete a lookup increases (see Figure 3). However, an interesting observation is that for larger topologies, the number of physical hops is in general lower when using Georoy as compared to Bamboo. This is because, due to the overlay addressing scheme in Georoy, the logical and physical topologies are tightly coupled so that the logical path does not differ much from the physical one. In fact, for large network topologies, the ratio between the physical and logical hops is around 2 for Georoy and rises to 5 for Bamboo. Since the formation of the overlay network is independent of the physical location of the nodes in Bamboo, for larger topologies the probability that a peer selects a close logical neighbor located far away in the physical topology is higher. This results in longer routes when topologies are larger. Also, note that the variance for the physical hops is much smaller in Georoy compared to Bamboo. This is again due to the addressing scheme of Bamboo, which randomly selects nodes in the overlay as neighbors, although they might be actually far away in terms of physical distance.

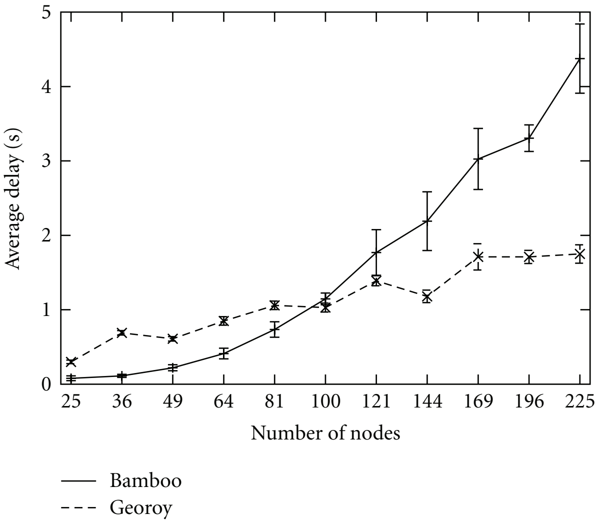

Figure 4: Comparison between the delay in Georoy and Bamboo in a grid topology.

In multihop wireless networks, the more hops a packet is forwarded, the larger the delay and, in general, the higher the packet loss probability. This is because at every intermediate node, the packet needs to compete for medium access and collisions due to, for example, hidden nodes might lead to frequent retransmissions and consequently high packet loss. The impact of an increase in the number of physical hops traveled in case of large topologies can be seen in the average lookup delay comparison shown in Figure 4. Here, we can see that for smaller topologies, Bamboo outperforms Georoy as less physical hops are required. However, due to the efficiency of its addressing scheme, the increase in the number of physical hops is smaller for larger topologies in Georoy, compared to Bamboo. Therefore, Georoy provides better lookup delays with larger topologies. Interestingly, Georoy shows smaller number of physical hops as compared to Bamboo when network size is larger than 144 nodes. However, the lookup delay of Bamboo is smaller as compared to Georoy already at a network size of about 100 nodes. This apparent discrepancy can be explained due to the fact that the random distribution of requests can turn into a different load on the links. There might be situations where the number of physical hops is a bit smaller for one protocol, but the load on the links might be different resulting in an advantage for the other protocol in terms of delay.

Another interesting observation is that the number of successfully completed lookups decreases as network size

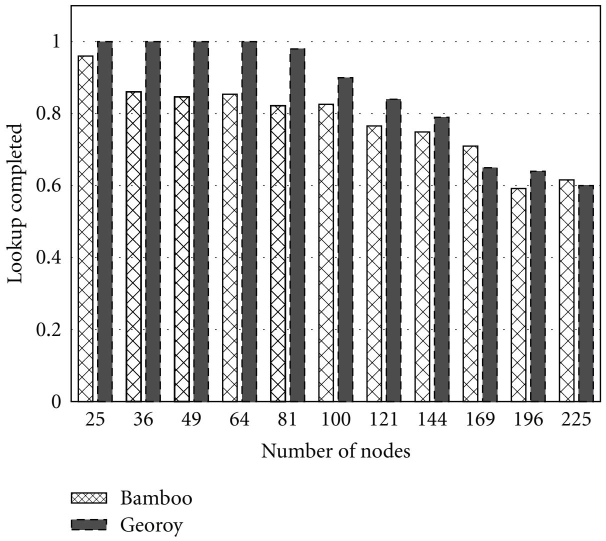

Figure 5: Comparison between the percentage of lookups completed in Georoy and Bamboo in a grid topology.

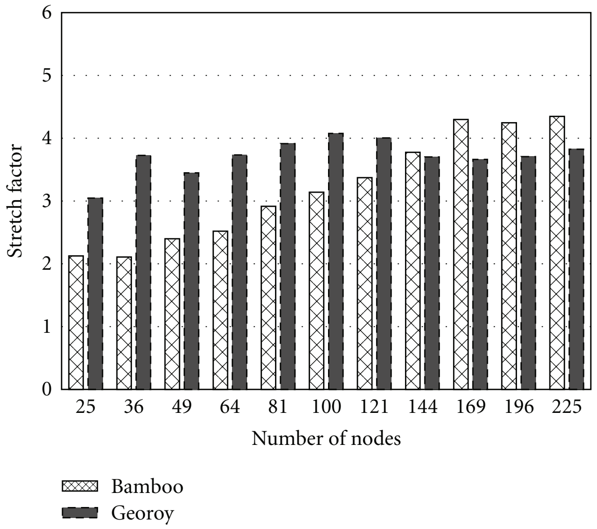

Figure 6: Comparison between the stretch factor in Georoy and Bamboo in a grid topology.

increases (see Figure 5). By increasing the number of nodes in the network we also increase the amount of messages exchanged (management traffic required to maintain the overlay plus key lookup request/replies) among the nodes and consequently the wireless contention for the medium. Also, when lookup packets traverse more hops, they need to compete more often for medium access and the probability to collide due to, for example, hidden nodes is higher. Interestingly, the number of completed lookup requests is smaller for Bamboo as compared to Georoy, even for small topologies. This can be attributed to the fact that the management traffic of Bamboo is significantly higher. Such high-management traffic leads to more load and contention leading to higher chance that the lookup request cannot be completed correctly [36]. In Bamboo, in this case the lookup request is retransmitted a limited amount of time until the agent gives up and declares the request as not successful.

The stretch factor presented in Figure 6 shows that both protocols can satisfy lookup requests with a limited increase in the number of hops traversed when compared to the shortest path approach. As we have seen in Figure 3, Georoy needs fewer hops to forward a lookup request to the destination when the network is composed of 144 nodes or more. Consequently, the stretch factor of Georoy is smaller compared to Bamboo at large network sizes.

When considering the random topologies, similar conclusions can be drawn. However observe that, in the random case, nodes are not distributed on the vertices of a grid, so physical proximity can help to reduce the number of physical hops and, thus, decrease delay significantly as evident in Figures 8 and 9. In fact when performing a lookup operation, one can move in any direction to a neighbor node which is not constrained to be located on a grid vertex. In addition, due to the random nature of the node location, we could observe more clustering of nodes as compared to a grid scenario. Therefore, as nodes are more close to each other in most of the area, less physical hops are required, thus implying less delay to complete the lookup operation. Clearly, due to the randomness in node location, there is more variability in the number of physical hops and delay. The logical hops instead do not vary much as compared to the grid scenario (see Figure 7).

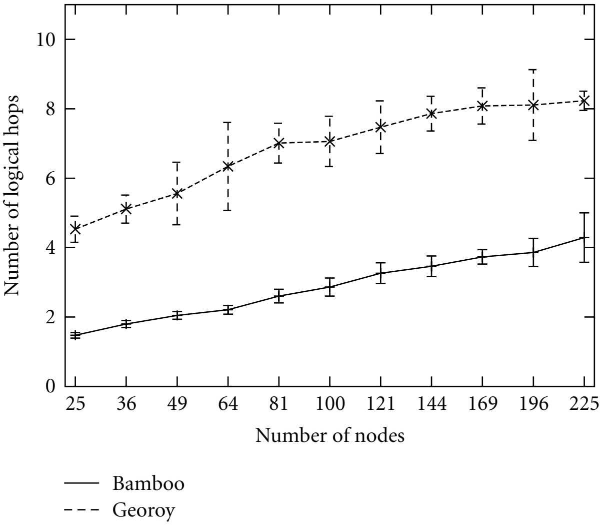

Figure 7: Comparison between the number of logical hops in Georoy and Bamboo for random topologies.

7.2. Impact of Number of Replication for Grid Topologies. Besides the impact of network size in grid and random topologies, another important point that we address is to determine the benefit of using a replication mechanisms in opportunistic scenarios. We start by looking at the impact of having different number of replicas as a way to speed up the resource lookup process. In our experiments we considered that both in Georoy and Bamboo each resource

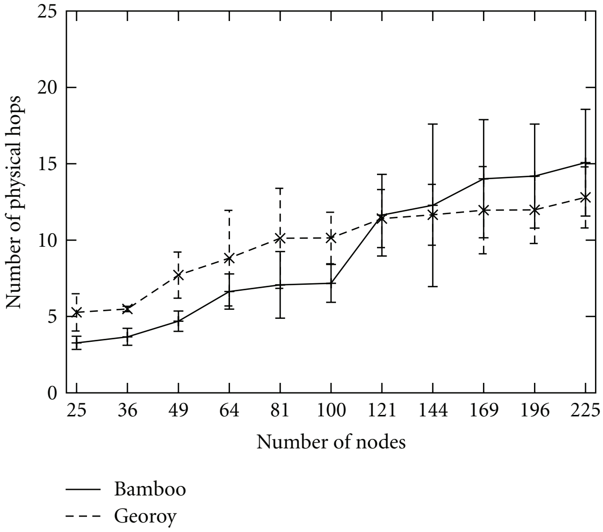

FIGURE 8: Comparison between the number of physical hops in Georoy and Bamboo for random topologies.

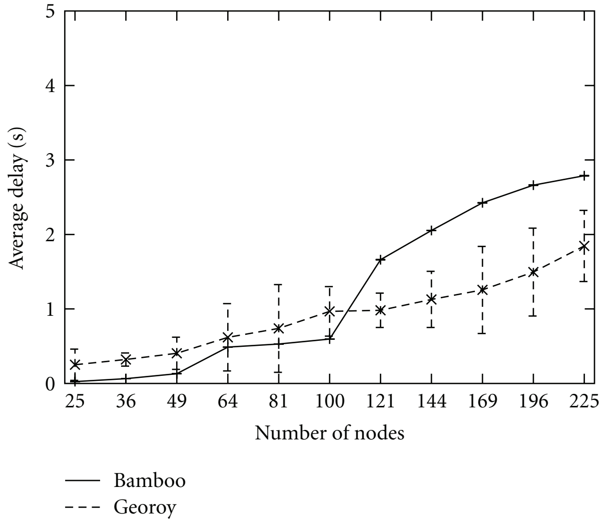

FIGURE 9: Comparison between the delay in Georoy and Bamboo for random topologies.

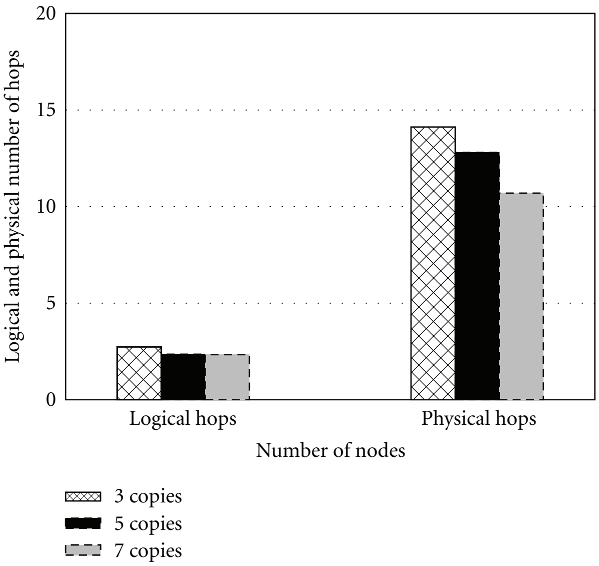

FIGURE 10: Number of logical and physical hops in Bamboo in a grid topology with 225 nodes.

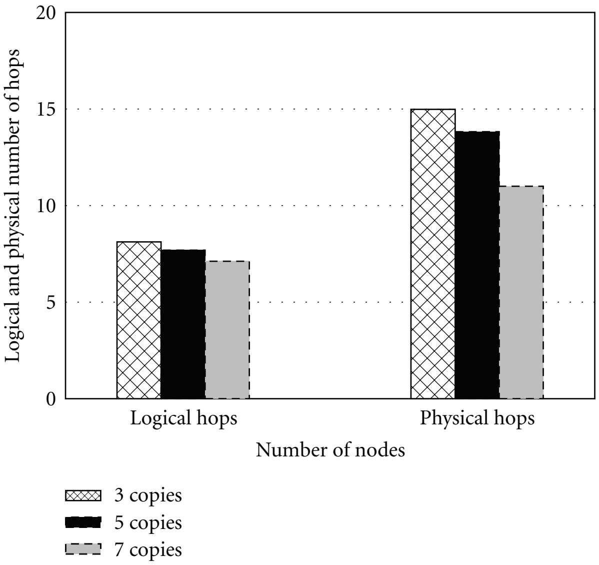

FIGURE 11: Number of logical and physical hops in Georoy in a grid topology with 225 nodes.

was replicated at 3, 5, or 7 different nodes. We assume a random waypoint mobility of the LP node providing the resource and consequently the replicas of the resource are randomly distributed in Georoy and are assigned to random nodes in the leafset in Bamboo, independently of the LP movement. In Figures 10 and 11 we observe that, upon increasing the number of copies of a resource, both the number of logical and physical hops slightly decrease. As expected this is because, when increasing the number of replicas, the probability of finding the resources closer raises as well. As a result, when using more replicas, the delay to complete a resource lookup can be reduced as evident looking at Figure 12. Also, consider that in Bamboo no significant variations in the number of logical hops as a consequence of a change in the number of resource replicas are met. The reason for this behavior is to be searched in the replication mechanism which in Bamboo disseminates replicas randomly at nodes in the leafset which are thus very close in the logical space but could not give meaningful help in speeding up the lookup procedure. Also observe that in Georoy it is sufficient to use a controlled number of replicas (i.e., higher than or equal to 5) to achieve quite stable performance.

7.3. Impact of Data Mule Mobility. Finally, we wanted to test how the two protocols behave in case of a disconnected scenario where an isolated node wants to perform a lookup

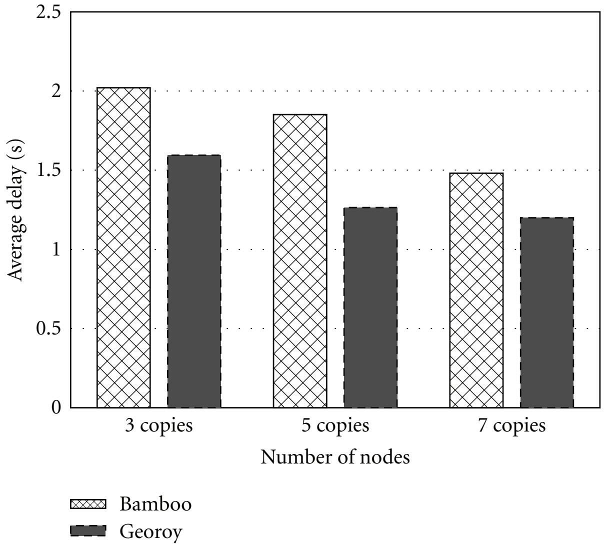

Figure 12: Delay in Georoy and Bamboo in a grid topology with 225 nodes.

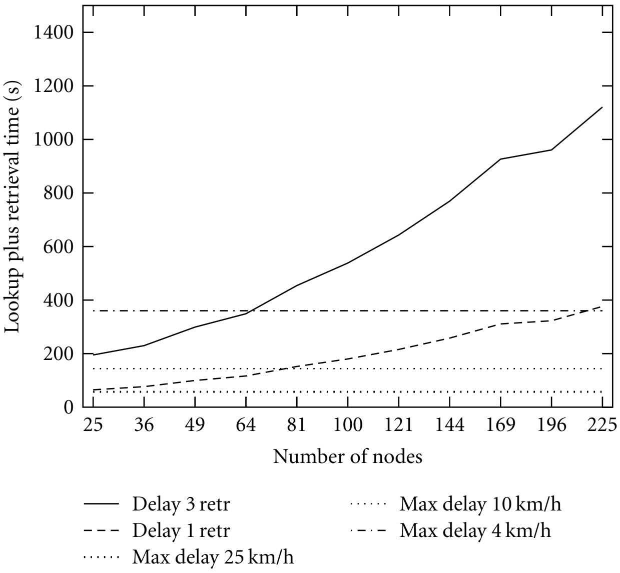

Figure 13: Delay-tolerant networking statistics in Bamboo.

but can only execute it during the limited time spent by a data mule, who travels around the network area, in its coverage range. In particular in this case we estimated the delay for a resource retrieval. We assumed a mobile data mule moving with a velocity variable in (4 Km/h (pedestrian case), 10 Km/h (vehicular case 1) and 25 Km/h (vehicular case 2)).

An isolated node, in the best case, will have the data mule in its coverage area for a time equal to $2 \cdot R/v$ where R is the transmission range and v is the data mule velocity. We assumed a retrieval for a file of size 2 MB with links of capacity equal to 1 Mbs. We considered a variable number of retransmissions on each link in [1, 3]. Accordingly in Figures 13 and 14, we show:

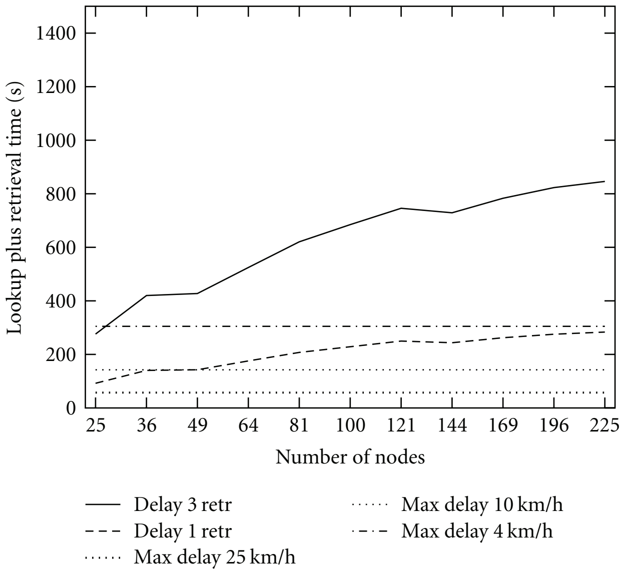

Figure 14: Delay-tolerant networking statistics in Georoy.

(i) the maximum delay taken for performing the lookup and retrieving the file in case of 1 retransmission on each link (delay 1 retr),

(ii) the maximum delay taken for performing the lookup and retrieving the file in case of 3 retransmissions on each link (delay 3 retr),

(iii) the maximum available time for lookup and retrieval depending on the data mule velocity (max delay).

Comparing the two plots we observe that for both Georoy and Bamboo the resources can be retrieved during the limited proximity time if the data mule moves around 4 Km/h. Instead, when the velocity of the mule is higher (10 or 25 Km/h), the percentage of retrieved resources during the contact time decreases and the delivery will be delayed of an amount equal to the intercontact time, that is, the time passed since previous exit until next entry of the mule into the coverage area of the isolated node. Supposing to employ a random way-point model for the data mule movements, the CDF of intercontact time [39] is shown in Figures 15 and 16 by varying the number of super peers and the mule's velocity, respectively.

Looking at the curves related to the velocity of mule around 10 Km/h, we observe that Georoy can complete the retrieval of a resource during the proximity time when the number of SPs is lower than or equal to 49; in Bamboo, instead, the delivery can be satisfied when the number of SPs is lower than 81.

Finally, in Figure 17 we show the percentage of downloads completed by an isolated node during the transit period of the data mule, when the latter moves at 10 Km/h, by varying the number of copies for each resource. As we observed, upon increasing the number of replicas, the retrieval procedure can be speed up: without replication, Georoy is able to efficiently exploit the limited proximity time for exchanging data until a maximum number of SPs

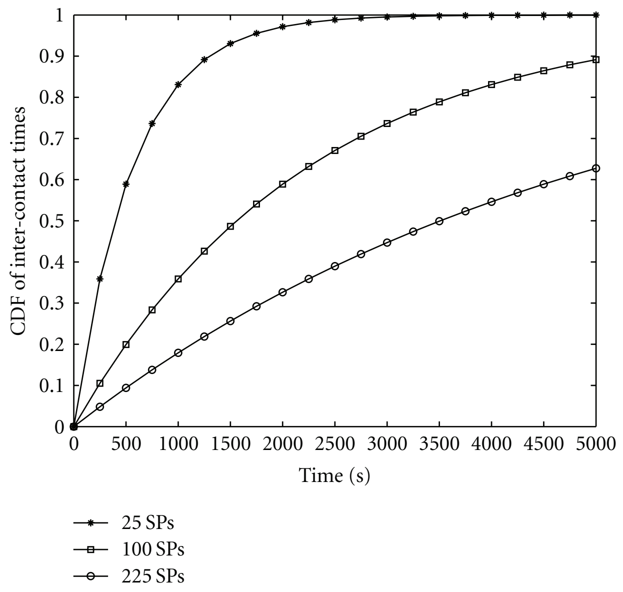

Figure 15: CDF of intercontact time for different number of SPs.

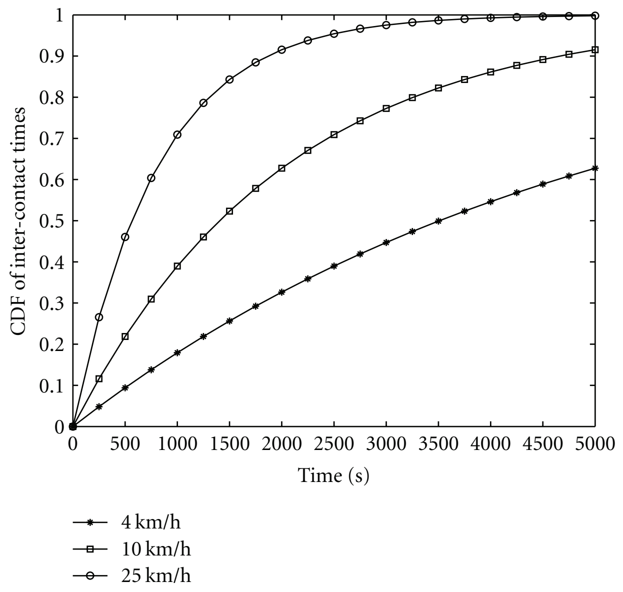

Figure 16: CDF of intercontact time for different data mule's velocity.

equal to 49; exploiting the random dissemination of the resources, instead, this threshold can be increased until 81 SPs employing 7 copies for each resource.

Conclusions of this analysis are the following.

(i) Bamboo performs better than Georoy in small to medium size topologies both grid or random. This is due to the more complete view of the overlay given by the larger overlay routing information, which also requires higher management traffic. When network size increases, Georoy overcomes Bamboo in performance due to the location aware addressing scheme.

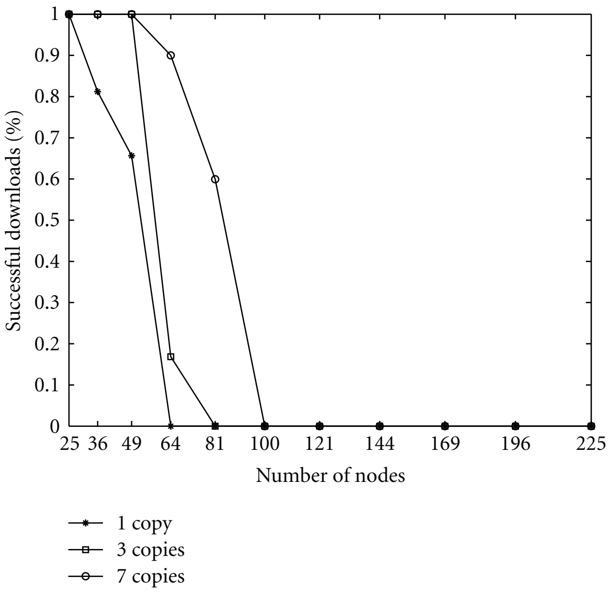

Figure 17: Replication effectiveness in resource downloading for Georoy.

(ii) Random topologies lead to a reduction in the number of hops and, thus, in the delay with respect to more regular cases like grid topology. This is mainly due to the clustering of nodes, which reduces the number of required physical hops.

(iii) Bamboo in general exhibits a lower number of lookups completed successfully due to its high overhead.

(iv) In opportunistic scenarios, where a data mule travels around and helps to connect remote nodes to infostations, when the data mule does not move too fast both protocols can allow lookup and delivery during the limited proximity time although Bamboo is more convenient also for slightly higher velocities. Performance improves when the download volume reduces or the data mule moves slower.

8. Conclusions

In this paper we addressed the problem of efficient content distribution and resource retrieval in opportunistic challenged scenarios. The latter are characterized by intermittent connectivity and, thus, use of traditional P2P approaches proposed for reliable and connected wireless networks does not always show effectiveness in these networks. Accordingly, we considered two efficient P2P schemes for wireless networks and enhanced them by introducing procedures to allow increasing scalability and reliability by use of multiple replicas of the same resource in the network and management of network disconnections. Performance results were aimed at comparing the performance of the two algorithms (Bamboo and Georoy) in both the case of static connected networks and delay-tolerant scenarios.

Our proposed extension focuses on scenarios where we have a set of infostations (SPs) which are connected through, for example, some backhaul wireless mesh network. In this case, the proposed techniques such as the replication strategy in Georoy together with the data mule concept, allow to improve performance with respect to the case of lack of replication. However, for a fully delay-tolerant networking scenario where no infrastructure is available and all nodes move around freely, the backhaul would be no longer connected all the time. Depending on the amount of connectivity, one can then question if such structured P2P approach would still be feasible for a fully disconnected DTN which would rather require physical contacts between SPs in order to exchange information. We argue that when only a few SPs are mobile, the structured P2P approach would still be feasible due to the redundancy of the wireless mesh backhaul, given enough replication is in place. When more and more SPs roam around leading to temporarily sparse deployments, the overlay structure will, at some point, no longer be maintainable and the protocols will not be able to cope with the harsh environment. In such case, epidemic information dissemination resorting to some form of broadcasting could lead to a better performance. However, at what point of mobility/sparse deployment structured P2P approaches fail to deliver suitable performance is out of the scope of the paper and should be related also to the specific application scenario being considered.

Acknowledgments

This work was partially supported by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (Contract no. 216715) and by the Italian National Project: "Wireless multiplatfOrm mimo active access netwoRks for QoS-demanding muLtimedia Delivery (WORLD)", under Grant no. 2007R989S.

References

[1] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile Ad Hoc Networks," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MoBiHoc '04)*, pp. 187–198, Tokyo, Japan, May 2004.

[2] SARI project, http://edev.media.mit.edu/SARI/.

[3] COW project, http://www.vidal.org.in/.

[4] J. James, "Mechanisms of access to the Internet in rural areas of developing countries," *Telematics and Informatics*, vol. 27, no. 4, pp. 370–376, 2010.

[5] B. Furuholt and E. Matotay, "Public internet access and E-government distribution in developing countries: evidence from Tanzania," in *Poceedings of the IFIP Workshop at Makerere University*, Kampala, Uganda, March 2010.

[6] Air Jaldi project, http://drupal.airjaldi.com/.

[7] I. F. Akyildiz, Ö. B. Akan, C. Chen, J. Fang, and W. Su, "InterPlaNetary Internet: state-of-the-art and research challenges," *Computer Networks*, vol. 43, no. 2, pp. 75–112, 2003.

[8] A. Rowstron and P. Druschel, "Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware '01)*, Heidelberg, Germany, November 2001.

[9] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling churn in a DHT," in *Proceedings of the Annual Technical Conference (USENIX '04)*, June 2004.

[10] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: a scalable and dynamic emulation of the butterfly," in *Proceedings of the 21st Annual ACM Symposium on Principles of Distributed Computing (PODC '02)*, pp. 183–192, Monterey, Calif, USA, July 2002.

[11] L. Galluccio, G. Morabito, S. Palazzo, M. Pellegrini, M. E. Renda, and P. Santi, "Georoy: a location-aware enhancement to Viceroy peer-to-peer algorithm," *Computer Networks*, vol. 51, no. 8, pp. 1998–2014, 2007.

[12] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup service for internet applications," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '01)*, pp. 149–160, San Diego, Calif, USA, August 2001.

[13] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X '02)*, pp. 96–107, October 2002.

[14] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proceedings of the ACM SIGCOMM Workshop on Delay Tolerant Networking and Related Topics (WDTN '05)*, pp. 244–251, Philadelphia, Pa, USA, August 2005.

[15] Reality Mining project, http://reality.media.mit.edu/publications.php.

[16] Haggle project, ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/fire/projects-haggle_en.pdf.

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04)*, pp. 235–244, October 2004.

[18] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, 2002.

[19] J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proceedings of the IEEE 61st Vehicular Technology Conference (VTC '05)*, pp. 2289–2293, June 2005.

[20] J. Leguay, T. Friedman, and V. Conan, "DTN routing in a mobility pattern space," in *Proceedings of the ACM SIGCOMM Workshop on Delay Tolerant Networking and Related Topics (WDTN '05)*, pp. 276–283, Philadelphia, Pa, USA, August 2005.

[21] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM '05)*, pp. 183–189, Los Alamitos, Calif, USA, June 2005.

[22] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: routing for vehicle-based disruption-tolerant networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, Barcelona, Spain, April 2006.

[23] K. Scott and S. Burleigh, "Bundle protocol specification," Tech. Rep., NASA Jet Propulsion Laboratory, November 2007.

[24] J. Seguí and E. Jennings, "Delay tolerant networking—bundle protocol simulation," in *Proceedings of the 2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT '06)*, pp. 235–240, July 2006.

[25] Y. Gong, Y. Xiong, Q. Zhang, Z. Zhang, W. Wang, and Z. Xu, "Anycast routing in delay tolerant networks," in *Proceedings of the Global Telecommunications Conference (GLOCOM '06)*, December 2006.

[26] W. Zhao, M. Ammar, and E. Zegura, "Multicasting in delay tolerant networks: semantic models and routing algorithms," in *Proceedings of the ACM SIGCOMM Workshop on Delay Tolerant Networking and Related Topics (WDTN '05)*, pp. 268–275, Philadelphia, Pa, USA, August 2005.

[27] Q. Ye, L. Cheng, M. C. Chuah, and B. D. Davison, "OS-multicast: on-demand situation-aware multicasting in disruption tolerant networks," in *Proceedings of the IEEE 63rd Vehicular Technology Conference (VTC '06)*, pp. 96–100, Melbourne, Australia, July 2006.

[28] K. Scott, "Disruption tolerant networking proxies for on-the-move tactical networks," in *Proceedings of the Military Communications Conference (MILCOM '05)*, October 2005.

[29] A. Balasubramanian, Y. Zhou, W. B. Croft, B. N. Levine, and A. Venkataramani, "Web search from a bus," in *Proceedings of the 2nd ACM Workshop on Challenged Networks (CHANT '07)*, pp. 59–66, September 2007.

[30] Napster, http://www.napster.com.

[31] Gnutella protocol specification v0.6, http://rfc-gnutella.sourceforge.net/.

[32] KaZaA, http://www.kazaa.com.

[33] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: a resilient global-scale overlay for service deployment," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 41–53, 2004.

[34] H. J. Siegel, "Interconnection networks for SIMD machines," *Computer*, vol. 12, no. 6, pp. 57–65, 1979.

[35] S. Rhea, B. Godfrey, B. Karp et al., "OpenDHT: a public DHT service and its uses," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 73–84, 2005.

[36] M. C. Castro, E. Villanueva, I. Ruiz, S. Sargento, and A. J. Kassler, "Performance evaluation of structured P2P over wireless multi-hop networks," in *Proceedings of the 2nd International Conference on Sensor Technologies and Applications (SENSORCOMM '08)*, pp. 796–801, August 2008.

[37] ns2, http://www.isi.edu/nsnam.

[38] Uppsala implementation of the AODV protocol, http://sourceforge.net/projects/aodvuu/files/AODV-UU/0.9.5/aodv-uu-0.9.5.tar.gz/download.

[39] M. Abdulla and R. Simon, "A simulation study of common mobility models for opportunistic networks," in *Proceedings of the 41st Annual Simulation Symposuim (ANSS '08)*, pp. 43–50, April 2008.