Mobile-relay Forwarding in Opportunistic Networks

Giuseppe Anastasi, **Marco Conti, Andrea Passarella, Luciana Pelusi**
Pervasive Computing & Networking Lab. (PerLab)
Dept. of Information Engineering
University of Pisa, Italy
`{firstname.lastname}@iet.unipi.it`
IIT-CNR
National Research Council, Italy
`{firstname.lastname}@iit.cnr.it`

1. Introduction

Opportunistic networks are one of the most interesting extensions of the legacy Mobile Ad hoc Networks (MANETs) concept. Legacy MANETs are composed by mobile nodes that collaboratively setup a network plane by running a given routing protocol. Therefore, the – sometimes implicit – assumption behind MANETs is that the network is well connected, and nodes' disconnection is an exception to deal with. Most notably, if the destination of a given message is not connected to the network when the message is generated, then that message is dropped after a short time (i.e., the destination is assumed not to exist). Opportunistic networks are mobile wireless networks in which the presence of a "continuous" path between a sender and a destination is not assumed. Sender and destination nodesmay never be connected to the network at the same time. The network is assumed to be highly dynamic, and the topology is thus extremely unstable and sometimes completely unpredictable. Nevertheless, the network must guarantee end-to-end delivery of messages despite frequent disconnections and partitions.

The opportunistic networking paradigm is particularly suitable to those environments which are characterized by frequent and persistent partitions. In Section 7 we will survey the most relevant case studies that rely on this paradigm. However, we can anticipate here a couple of example scenarios where classical wireless networking approaches are not feasible and the opportunistic networking approach is the only viable solution. In the field of wildlife tracking, for example, some kinds of sensor nodes are used to monitor wild species. In these cases it is not easy (nor possible sometimes) to have connectivity among a source sensor node and a destination data collector node. This because the animals to be monitored move freely and there is no possibility to control them in such a way to favor connectivity. Opportunistic networks may also be exploited to bridge the digital divide. In fact, they can support intermittent connectivity to the Internet for underdeveloped or isolated regions. This can be obtained by exploiting mobile nodes that collect information to upload to the Internet as well as requests for web pages or any kind of data that needs to be downloaded from the Internet. Both data and requests are up/down-loaded from/to the Internet once the mobile data collector node reaches a location where connectivity is available.

It clearly emerges that routing and forwarding play a key role in opportunistic networks. However, given the intermittent connectivity, it is not always possible to define a complete route between the source and destination nodes at the moment the source is willing to deliver its message. Hence, routing is not intended in the classical way. Routes in opportunistic networks are usually computed "on-the-fly", while messages are being forwarded. Routing is thus rather concerned with finding hop-by-hop a path to the destination. In fact, at each step the only decision which can be made is to whom the message is to be forwarded next. As a result, routing and forwarding are typically performed at the same time[1].

In general, two main concepts are at the basis of routing/forwarding protocols for these networks. On the one hand, since topological information is unreliable, routing should exploit information pertaining to any layer of the stack to understand how to build routes. On the other hand, any communication opportunity should be exploited (at least, considered) for carrying messages

---

[1] In the following, we will use the terms routing and forwarding interchangeably.

closer to the eventual destination(s).

Different approaches to routing are possible, as discussed in Section 2. Some (historical) routing approaches are based on a vast dissemination of data in all network directions. By spreading information throughout the network, the probability that messages eventually reach destination is very high. However, these approaches cause severe resource consumption (e.g., bandwidth and memory at intermediate nodes) due to the frequent data exchanges involved. More recent approaches therefore, tend to identify only one or a few preferential directions for data forwarding. These approaches are generally more computationally intensive than the previous ones, but consume smaller amount of bandwidth and memory. Although we provide here a general classification and discussion of the possible routing approaches in opportunistic networks, we particularly focus on Mobile-Relay Forwarding (MRF). MRF assumes that there exist particular nodes (Mobile Relays) in the network that are exploited to collect messages from the source nodes, and to take messages (closer) to the destination. Routing approaches based on Mobile Relays (MRs) are very energy efficient because regular nodes are relieved of their routing workload, which is instead undertaken by MRs. Furthermore, this approach increases network scalability since the addition of extra nodes to the network does not imply an increment of routing complexity. This is particularly beneficial to scenarios that can potentially include a lot of (heterogeneous) devices like, for example, an urban environment. For example, MRs can be bus traveling in a city, while regular nodes can be pedestrians. Regular nodes wait for one of such MRs to pass nearby, and hand over messages to it. Usually, MRs have completely different mobility patterns with respect to regular nodes, cover larger distances, and are thus able to connect nodes that would not be able to communicate otherwise. Mobility of MRs can be either controllable or not, and MRs can be either already part of the system, or deployed just for the sake of improving routing performance. Furthermore, MRs usually have fewer restrictions on the availability of resources with respect to the other nodes of the network. Therefore, they can greatly increase network connectivity and data delivery in opportunistic networks.

In this chapter we describe the different types of architecture that have been proposed to exploit the MRF concept (see Section 3). Then, we identify the main issues to be addressed in the design of the MR behavior (Section 4), and focus on algorithms that control MR movements to optimize network performance (Section 5). As opportunistic networks are composed by mobile devices, power-conservation issues should be of primary concern. Therefore, we discuss power-management techniques for MRF in Section 6. Section 7 is devoted to the description of some relevant case studies highlighting how the MRF concept can be exploited in different scenarios. In particular, in Section 7.4 we give special emphasis to a novel kind of network that the MRF concepts can be applied to, i.e., Underwater Sensor Networks. Conclusions and open issues are discussed in Section 8.

2. Routing Approaches in Opportunistic Networking

As highlighted above, routing is the most compelling challenge in opportunistic networking. The design of efficient routing strategies for opportunistic networks is generally a complicated task due to the absence of knowledge about the topological evolution of the network. Routing performance improves when more knowledge about the expected topology of the network can be exploited [Sush04]. Unfortunately, this kind of knowledge is not easily available, and a trade-off must be met between performance and knowledge requirement.

Depending on the particular routing approach followed, opportunistic networks may be classified at a very top level into two categories: *infrastructure-less* and *infrastructure-based* networks [Pelu06b]. Infrastructure-less networks are completely flat ad hoc networks (*without* *infrastructure*) where all the nodes equally take on routing/forwarding duties. In infrastructure-based networks (*with infrastructure*) instead, some form of infrastructure is exploited to forward messages opportunistically. The infrastructure is typically composed by special nodes that are in charge of messages forwarding whereas the other nodes are generally relieved of the forwarding workload.

2.1. *Infrastructure-less Opportunistic Networks*

In infrastructure-less opportunistic networks two basic routing approaches are followed: *dissemination*-based and *context*-based routing. Dissemination-based algorithms are essentially forms of controlled flooding, and differentiate themselves for the policy used to limit flooding. Context-based approaches usually do not adopt flooding schemes, but use knowledge of the context that nodes are operating in to identify the best next hop at each forwarding step. The following sub-sections offer an overview of both dissemination-based and context-based routing approaches describing the most representative algorithms of each.

2.1.1 *Dissemination-based Routing*

Routing techniques based on data dissemination perform delivery of a message to destination by simply diffusing it all over the network. The heuristic behind this policy is that, since there is no knowledge of a possible path towards the destination nor of an appropriate next-hop node, a message should be sent everywhere. It will eventually reach the destination by passing node by node. Dissemination-based techniques are very resource hungry. Moreover, due to the considerably high number of transmissions involved, dissemination-based techniques suffer from high contention and may potentially lead to network congestion. To increase the network capacity the spreading radius of a message is typically limited by imposing a maximum number of relaying hops to each message or even by limiting the total number of message copies present in the network at the same time. When no relaying is further allowed, a node can only send directly to destination when/in case met.

The first protocol exploiting dissemination techniques that has been proposed in the literature is the *Epidemic Routing* protocol [Vahd00]. In Epidemic Routing messages diffuse in the network similarly to diseases or viruses, i.e., by means of pair-wise contacts between individuals/nodes. A node is *infected* by a message when it either generates that message or alternatively receives it from another node for forwarding. The infected node stores the message in a local buffer. A node is *susceptible* to infection when it has not yet received the message[2] but can potentially receive it in case it comes into contact with an infected node (i.e., a node that stores that message). The infected node becomes *recovered* (healed from the disease) once having delivered the message to the destination node and, as a result, it also becomes *immune* to the same disease and does not provide relaying to the same message any more. The dissemination process is somehow bounded because each message when generated is assigned a *hop count limit* giving the maximum number of hops that that message is allowed to traverse till the destination. When the hop count limit is one, the message can only be sent directly to the destination node.

Further steps beyond epidemic routing are represented by *PROPHET* [Lind03] and the *MV routing* [Burn05] protocols. In both protocols, messages are exchanged during pair-wise contacts as in epidemic routing. However, a more sophisticated method to select the messages to forward to an encountered node is introduced. Basically, the choice depends on the *probability* of the encountered nodes to deliver the messages successfully to their eventual destinations. The delivery probability relies on observations on the *meetings* between nodes (in PROPHET), and both on the *meetings* between nodes and the *visits* of nodes to geographical locations occurred in

---

[2] The message itself represents the infection/virus.

the recent past (in MV Routing).

Network-coding-based routing [Widm05] also falls in the category of dissemination-based algorithms, but takes an original approach to limit message flooding. Messages are combined together (encoded) at nodes before being forwarded. Then, the codes produced are sent out instead of the original messages. Codes are spread in different directions like in other dissemination-based routing protocols. The number of codes generated is higher than the number or original messages combined together. This is to allow much more robustness against both packet and path loss. Encoding is performed at both source and intermediate nodes.

Just to give a classical, and simplified, example of network coding, let A, B, and C, be the only three nodes of a small network (see Figure 1). Let node A generate the information "*a*" and node C generate the information "*c*". Then, suppose the information produced needs to be known at all the nodes. Hence, node A and node C send their information to node B, then node B rather than sending two different packets for "*a*" and "*c*" respectively, broadcasts a single packet containing "*a*" *xor* "*c*". Once received "*a*" *xor* "*c*", both nodes A and C can finally infer the missing information (i.e., node A can infer "c" and node C can infer "a"). Network-coding-based routing can be generalized by recursively using erasure-coding techniques at intermediate nodes [Pelu06a]. It outperforms flooding, as it is able to deliver the same information with a fewer number of messages injected into the network.

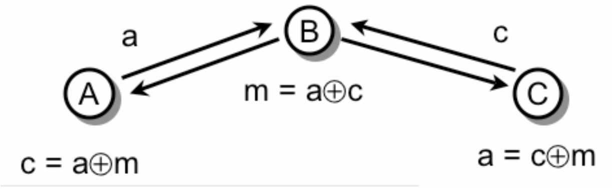

Figure 1. Example of network-coding efficiency.

2.1.2 *Context-based Routing*

Most of the dissemination-based techniques limit messages' flooding by exploiting knowledge about direct contact with destination nodes. Context-based routing exploits more information about the context nodes are operating in to identify suitable next hops towards the eventual destinations. The usefulness of a host as next hop for a message is hereafter referred to as *utility* of that host. Usually, such routing techniques are able to reduce significantly the message duplication and resource consumption (e.g., bandwidth, memory, energy) of dissemination-based techniques. Since they also reduce network congestion, it has been shown that they are able to reduce delays and message loss as well. The main cost paid for these advantages is the fact that context information must be kept at nodes and circulated among nodes. However, recent results show that resource consumption is far lower even when these additional costs are considered [Bold07a].

In the *Context-Aware Routing (CAR)* protocol [Muso05] each node in the network is in charge of producing its own delivery probabilities towards each known destination host. Delivery probabilities are exchanged periodically so that, eventually, each node can compute the best carrier for each destination node. The best carriers are computed based on the nodes' context. Among the context attributes needed to elect the best carrier there are, for example, the residual battery level, the rate of change of connectivity, the probability of being within reach of the destination, the degree of mobility. When the best carrier receives a message for forwarding, it stores it in a local buffer and eventually forwards it to the destination node when met, or alternatively to another node with a higher delivery probability. Actually, CAR provides a framework for computing next hops in opportunistic networks based on the *multi-attribute utility theory* applied to generic context attributes. Simulation results show that CAR is more scalable than epidemic routing as the protocol overhead is approximately constant regardless of the node buffer size.

In *MobySpace Routing* [Legu06] the nodes' mobility pattern represents the context information used for routing. The protocol builds up a high dimensional Euclidean space, named *MobySpace*, where each axis represents a possible contact between a couple of nodes and the distance along an axis measures the probability of that contact to occur. Two nodes that have similar sets of contacts and that experience those contacts with similar frequencies are close in the MobySpace. The best forwarding node for a message is the node that is as close as possible to the destination in this space. This in fact improves the probability that the message will eventually reach the destination. Obviously, in this *virtual contact space* just described, the knowledge of all the axes of the space also requires the knowledge of all the nodes that are circulating in the space[3].

Both CAR and MobySpace Routing require full knowledge of possible destinations to enable forwarding. The History-Based Opportunistic Routing Protocol (HiBOP) [Bold07a, Bold07b] provides a framework for managing and exploiting context information that does not require all nodes to know each other. In HiBOP nodes exchange context information about the users when getting in touch. Each node remembers context information seen in the past (such information is enforced based on how often it is "seen" on encountered nodes). A node carrying a given message asks the encountered nodes to compute their delivery probability towards the destination(s). The delivery probability is computed based on the match between context information *about the destination* stored in the message itself, and context information stored by the encountered node itself. Messages are forwarded along a gradient defined by increasing match between the destination information and the context information of the carrying node. Hence, the algorithm dynamically selects as next hops those nodes that share more and more context information with the destination(s). HiBOP exploits social relationships among users to

---

[3] [Legu06] also proposes an optimization that does not require the knowledge of all contacts between nodes.

identify good carriers for messages.

2.2. *Infrastructure-based Opportunistic Networks*

Infrastructure-based opportunistic networks are characterized by the presence of special nodes that are used for collecting messages from source nodes and delivering them to their destinations. Such special nodes are generally more powerful than regular nodes as they have high energy budget and large storage capacity. They may either act as a gateway toward a less challenged network (e.g., the Internet), or they can simply increase the connectivity between (regular) nodes in the network. Depending on the mobility of special nodes, we can distinguish opportunistic networks with *fixed infrastructure* and with *mobile infrastructure*, respectively. When using a *fixed infrastructure* special nodes are stationary and are located at specific geographical points. On the other hand, in opportunistic networks with *mobile infrastructure* special nodes move around in the network area following either pre-defined or completely random paths.

2.2.1 *Routing based on Fixed Infrastructure*

A fixed infrastructure consists of special fixed nodes, i.e., base stations, which are sparsely deployed all over the network and act as message collectors. Base stations offer high capacity and robust data exchanges to the mobile nodes nearby. Moreover, they have high storage capacity to collect data from many nodes passing by. A source node wishing to deliver a message keeps it until it comes within reach of a base station, then forwards the message to the base station.

Base stations are generally gateways towards less challenged networks (e.g., they can provide Internet access or be connected to a LAN). Hence, the goal of an opportunistic routing algorithm is to deliver messages to the gateways, which are supposed to be able to find the eventual destination more easily. Two variations of the protocol are possible. The first one works exactly as described above, and only node-to-base-station communications are allowed. As a result, messages experience fairly high delays. The classical example of this approach is the Infostation model [Good97]. A second version of the protocol allows both node-to-base-station and node-to-node communications. This means that a node wishing to send a message to a destination node delivers the message to the base station directly if within communication range, otherwise it delivers the message opportunistically to a near node that will eventually forward it to the base station when encountered (routing schemes presented earlier can be used in this phase). Such a protocol has actually been proposed in the Shared Wireless Infostation Model (SWIM) [Smal03]. As it results from the above examples, historically, fixed base stations play a passive role in the opportunistic forwarding strategy because they simply act as information sinks (e.g., Infostations [Good97]). However, many benefits can be envisioned by running an opportunistic routing algorithm also at base stations. Base stations, for example, can simply collect the messages sent by the visiting nodes and then wait for the destination nodes to be within reach to forward the stored messages to them. Base stations of a mobile infrastructure (described in the next section) typically play such an active role.

Despite allowing energy saving at the mobile nodes (which are relieved of the forwarding workload, at least in the first version of the protocol), a routing approach relying on a fixed infrastructure is highly expensive due to the costs of the infrastructure. Moreover, it suffers from scalability issues since the addition of new nodes implies the expansion of the infrastructure. Using a mobile infrastructure instead of a fixed infrastructure is a valuable opportunity to realize a cheap and flexible infrastructure. A mobile infrastructure is composed of mobile nodes that move around in the network place following either pre-determined or arbitrary routes and gather messages from the nodes they approach. These special nodes may be referred to as *carriers*, *supports*, *forwarders*, *MULEs*, or even *ferries*. They can be the only entities responsible for the delivery of messages, when only node-to-carrier communications are allowed, or they can simply help increase connectivity in sparse networks and guarantee reachability of isolated nodes. In the latter case, delivery of messages is accomplished by both carriers and ordinary nodes and communications are allowed both node-to-node and node-to-carrier.

3. Forwarding Architectures for Opportunistic Networks with Mobile Relays

As mentioned in the Introduction, the rest of this chapter is focused on opportunistic networks with MRs. Therefore, in this section we start the discussion by presenting the possible kinds of architecture for this approach.

Figure 2 shows the system architecture of opportunistic networking with MRs. We can distinguish the following three different components: regular nodes, MRs, and base stations.

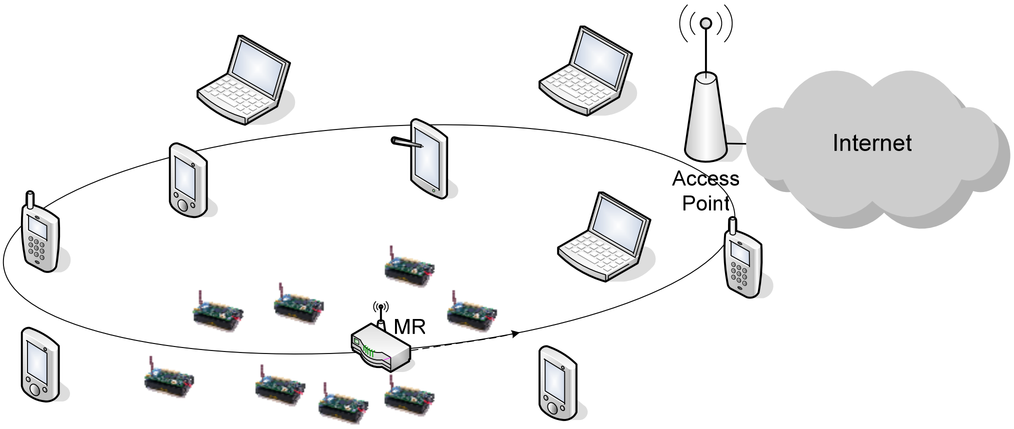

Figure 2. System Architecture for Opportunistic Networking with Mobile Relays.

Regular Nodes (or simply nodes, for short) are the information sources and destinations. Depending on the specific application scenario, they may be fixed or mobile. For instance, in a sensor network nodes are typically stationary, while in a Mobile Ad Hoc Network (MANET) they are usually mobile.

Mobile Relays (MRs) are specialized nodes that move throughout the network to collect data from source nodes and deliver it to the destination node or the Access Point. They can follow a fixed or variable trajectory, at constant or variable speed. Therefore, the time interval between successive visits of an MR to the same node may be predictable, variable in a bounded range, or completely random. The number of MRs in a network may vary depending on several factors such as, number of regular nodes, amount of traffic to manage, Quality of Service (QoS) requirements, and costs.

Access Points (APs) are infrastructured nodes serving as gateways towards less challenged networks (e.g., they provide connectivity to the Internet or a LAN). Again, the number of APs can vary depending on the number of nodes, number of MRs, traffic load, QoS requirements, installation costs, and so on.

When designing an opportunistic networking system based on the above architecture the following design issues need to be taken into account [Zhao03, Kans04].

- *Node Mobility*. Regular nodes may be stationary or mobile, depending on the application scenario. In case of mobile nodes we can distinguish between *task-driven* and *message-driven* mobility. In task-driven mobility nodes move according to a path that is dictated by a specific task or goal (e.g., a person with a PDA moves to go to work). In message-driven mobility node movements are aimed at data transmission/reception in general (e.g., a person with a PDA moves towards an MR to exchange messages with it).
- *Coordination between nodes*. Typically there are many nodes in the network, densely or sparsely deployed, depending on the application requirements. Of course, nodes can communicate with an MR only when it is within their communication range. Therefore, either nodes are mobile or they must be deployed at a distance from the MR trajectory not greater than their communication range. Alternatively, nodes can organize themselves to

form *clusters* [Soma06] or *regions* [Harr06]. Each cluster consists of a set of nodes that elect a specific node to act as a *gateway* node in charge of communication with the MR(s). Nodes in the cluster send their messages to the gateway by *multi-hop communication*, and the gateway transmits such messages to the MR(s) (see Figure 3). The same approach is used for message reception.

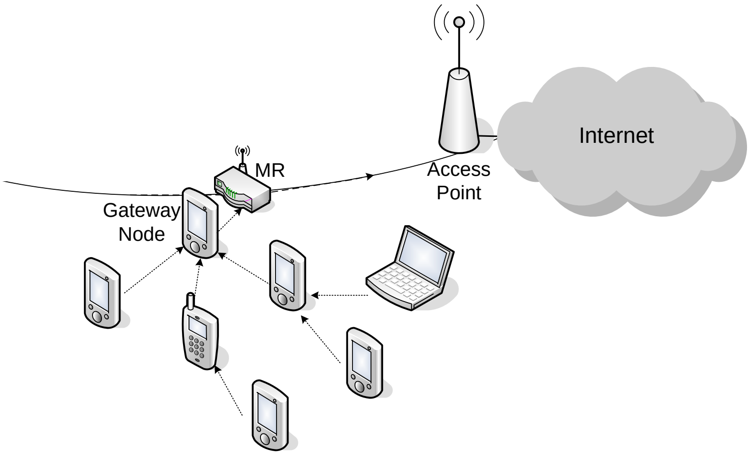

Figure 3. Opportunistic access through a gateway node via multi-hop communication.

- *MR mobility*. The mobility of MRs is a critical factor since it directly impacts the success of message delivery as well as the latency experienced by messages. To allow good connectivity among nodes MRs should be able to approach as many nodes as possible and to visit them with an appropriate frequency. In practice, they may be either part of the environment (e.g., a bus or person), or intentionally included as part of the network infrastructure (e.g., a mobile robot). In the former case there is typically no control on the MR mobility. In the latter case it is usually possible to control the MR trajectory and/or speed.
- *MR designation*. MRs may be either special nodes specifically designed to act as MRs, or regular nodes that serve temporarily as MRs. In the former case MRs are typically resource-

rich devices. In the latter case they have limited resources like all the other nodes in the system. In addition, an algorithm for designating MRs is required.

- *Number of MRs*. An opportunistic networking system may rely on one or more MRs, depending on performance, scalability and reliability requirements. Obviously, the capacity of a single MR is limited by its movement capability. Increasing the number of MRs allows for increased scalability and geographic coverage. In addition, a system with multiple MRs is more resilient to MR failures. On the other hand, a larger number of MRs implies higher economical costs. Therefore, the optimal number of MRs must be traded off between performance and costs.
- *Coordination between MRs*. If there are multiple MRs in the system, they may have similar or different capabilities. Furthermore, they may operate independently of each other, or in cooperation. In the latter case, a message can be exchanged between several MRs before reaching the destination node or an Access Point.
- *MR trajectory*. The trajectory followed by an MR to visit nodes may be fixed or variable. In the latter case it is adjusted dynamically depending on nodes requests, messages deadlines, etc. Obviously, this is possible only when an MR is part of the system infrastructure and can thus be controlled. In case of multiple MRs, a problem related to the MR trajectory design is the assignment of nodes to the MRs. The trajectory design should take into account not only routing but also load balancing among MRs.
- *MR Speed.* MRs may move at a constant, variable, or controlled speed. In the latter case the speed can be controlled by the MR software and adjusted dynamically to improve the communication performance. Again, this is possible only when MRs are part of the system infrastructure.

- *Power management and MR discovery*. As nodes have limited energetic resources they should switch their radio in sleep (i.e., low-power) mode when they are not involved in communications with MRs. However, since MR arrivals are usually unpredictable, this may prevent nodes from discovering an incoming MR. Energy-efficient discovery schemes are thus required that minimize energy consumption while keeping the probability of missing contacts with MRs as low as possible.
- *Data Collection and Delivery*. A message generated at a source node requires several communications to reach the destination node or an Access Point. The message is first transmitted by the source (or gateway) node to an MR. In case of multiple coordinated MRs, the message may be exchanged between several MRs before delivery to the destination node or Access Point. Protocols for efficient node-to-MR, MR-to-MR, and MR-to-AP communications are thus required.

In the following sections we will describe how the most relevant of the above issues have been addressed in practical opportunistic systems. Specifically, we will discuss the impact of different MR mobility patterns on forwarding, and also the related power-management issues.

Before going on, we conclude this section with some comments about the data collection and delivery process. The most interesting aspect of this problem is managing communications between MRs and regular nodes (or the gateway node when a clustering approach is used). Communications between the nodes of a cluster and the related gateway usually borrow well-known techniques from the MANET literature (such as clustering), while communications between MRs and Access Points are not particularly challenging.

Authors in [Kans04, Soma06] use a *stop-and-wait* protocol for communication between the MR and a regular node (or the gateway node of a cluster). MR sends an acknowledgement to the sending node for each message correctly received. The node transmits the next message only after receiving acknowledgement from the MR. If the acknowledgement is not received within a predefined *timeout* the node retransmits the message. The node starts transmitting data as soon as it discovers the MR in its proximity. No information about the location of the MR is exploited because such information may not be available in all systems. In [Anas07] it is analytically shown that using a window-based scheme with a window size greater than one provides a higher throughput and, for a fixed amount of data to transfer, it also lets the transfer time (and, hence, the energy consumption) decrease. However, increasing the window size beyond a given threshold may be unpractical since the MR could move out of the communication range. This would result in useless message transmissions (and energy consumption).

4. Mobile Relays

As anticipated in Section 3, MRs may be classified in two broad categories: they can be *part of the environment*, or specifically designed *as part of the network infrastructure*. Depending on their nature they may have different mobility patterns, as shown in Figure 4. When the MR is part of the environment its mobility is driven by the specific task the mobile element acting as MR is intended for, and cannot be controlled in any way. Conversely, when the MR is part of the system, its mobility can be controlled to improve the communication performance and extend the geographic coverage. However, even when the MR is not controllable, it may have different mobility patterns. If it follows a strict schedule it has a completely predictable mobility (e.g., a shuttle for public transportation). On the opposite side, it may have a completely random behavior so that no reliable assumption can be made on its mobility. Finally, the MR may follow a mobility pattern that is neither predictable nor completely random. For example, this is the case of a public transportation bus, or a car, that moves in a city and whose speed is subject to large variation due to traffic conditions. In such a case, the MR mobility pattern, even if not predictable, can be learned based on successive observations and estimated with some accuracy. Learning the MR mobility pattern and estimating times between successive MR visits to the same node is very important to save energy at the node, as it will be shown in Section 6.

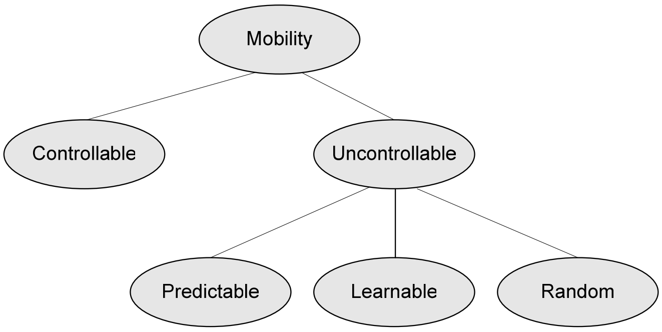

Figure 4. Classification of MR mobility.

A lot of examples of opportunistic networking systems have been proposed in the literature where MRs have different nature and mobility patterns.

Shah et al. [Shah03] propose a three-tier architecture for energy-efficient data collection in sparse sensor networks based on data MULEs (Mobile Ubiquitous LAN Extensions). Data MULEs can be people, animals, cars, buses, etc, passing nearby sensor nodes and collecting data from them. Obviously, as they are part of the environment, applying the MR forwarding concept to this scenario comes for free. On the other hand, they typically move randomly and no control is possible on them. An approach similar to data MULEs is exploited in the Predictable Mobility Architecture described in [Chak03]. The authors rely on a public transportation shuttle for data collection in sensor networks inside a campus. Unlike data MULEs, the shuttle is assumed to have a strict schedule and predictable inter-visit times. This greatly helps optimize power consumption at nodes. Public transportation buses are also the mobile elements in the Ad Hoc City project [Jetc03]. This project is aimed at creating a city ad hoc network where the role of mobile routers is played by public buses.

Both the Zebranet [Juan02] and SWIM [Smal03] projects focus on tracking wild species, and use MRs. In Zebranet animals to be tracked are zebras wearing special collars. The MR consists of a vehicle that periodically moves around in the savanna and collects data from the encountered zebras. Zebras collect data from other zebras and deliver it to the MR, thus they also act as MRs. In the Shared Wireless Infostation Model (SWIM) [Smal03] special tags are applied to whales to perform periodic data monitoring. Data is diffused at each pair-wise contact between whales and finally arrives to special SWIM stations that can be fixed (on buoys) or mobile (on seabirds). Hence, both whale-to-whale and whale-to-SWIM station communications are allowed and the MRs consist of both the mobile SWIM stations and the whales themselves. From the SWIM stations data is eventually forwarded to an Access Point on shore from where it will be finally delivered to destination for processing and utilization.

Many opportunistic networking systems use *controllable* MRs. Among the most relevant there are, for example, Message Ferrying [Zhao03, Zhao04], Inter-Regional Messenger [Harr06], and Controllably Mobile Infrastrucure [Kans04]. The main features of these systems are described in the next section.

Finally, there are several proposals (mainly targeted to sensor networks) that do not rely on MRs for transporting data mechanically from nodes to the Access Point. Instead, the Access Point itself is mobile and can change its position from time to time [Gand03, Akka05, LuoH05, Wang05]. This may be beneficial in terms of energy saving and decreased message latency. For example, moving the Access Point close to an area of heavy traffic or near loaded nodes helps reduce the total transmission power and extend the lifetime of nodes on the path of heavy traffic. [Akka05]. However, this scenario is beyond the scope of this chapter.

5. Motion Control

MR motion control can be achieved only when using a controllable mobile element. Motion control can be performed along two orthogonal directions: space and time. In the space direction we can define, and adapt dynamically, the *trajectory* followed by the MR to visit nodes. In the time direction, we can control the MR *speed* and adjust it to improve the communication performance. Another valuable form of control in opportunistic environments is topology control. In this case, nodes may decide to increase (or decrease) the *transmit power* to increase (decrease) contact times.The three different forms of motion control are discussed in the next subsections.

5.1. *Trajectory Control*

Figure 5 shows a classification of trajectory control approaches. We can distinguish them into two broad categories depending on whether the path followed by the MR to visit nodes is *fixed* or *variable*.

When the MR trajectory is *fixed* there is actually no control. Therefore, the trajectory must be defined very carefully especially when nodes are stationary. In particular, the following requirements need to be fulfilled. First, the MR trajectory should pass close to nodes, at a maximum distance less than the node communication range. Second, the MR should remain within the node communication range for enough time to allow a complete data exchange. Finally, the node trajectory must be feasible, i.e., compatible with geographical constraints. If the MR is a controllable mobile element (e.g., a mobile robot) its trajectory can be designed so as to address all the above requirements [Soma06, Zhao03]. If the MR is part of the environment (e.g., a bus), its trajectory depends on the specific task it carries out (e.g., public transportation), and cannot be changed. Hence, nodes must be deployed along the MR trajectory at a distance less than or equal to the node communication range [Jain06, Chak03, JeaS05]. The MR trajectory becomes less critical when nodes are mobile as they can move towards the MR when they want to exchange data [Zhao04].

Trajectory control actually consists in adjusting the MR trajectory dynamically, based on node requirements As shown in Figure 5 trajectory control techniques can be classified into two main categories: *on-demand* and *priority-based* techniques. When using an on-demand approach, the MR adapts its movements to satisfy nodes' requests. Each time a node has data to exchange, it sends a service request to the MR by using a long range radio, and the MR modifies its route to approach the requesting node. Nodes may be fixed or mobile [Zhao04]. Priority-based route control is used when nodes have different characteristics in terms of message generation rate and/or buffer size. Therefore, they need to be visited with different frequencies. In practice, each node is associated with a deadline, defined as the time when the node buffer will overflow. Nodes' deadlines are used to schedule the MR visits to nodes [Soma04, GuBo05,GuBo06].

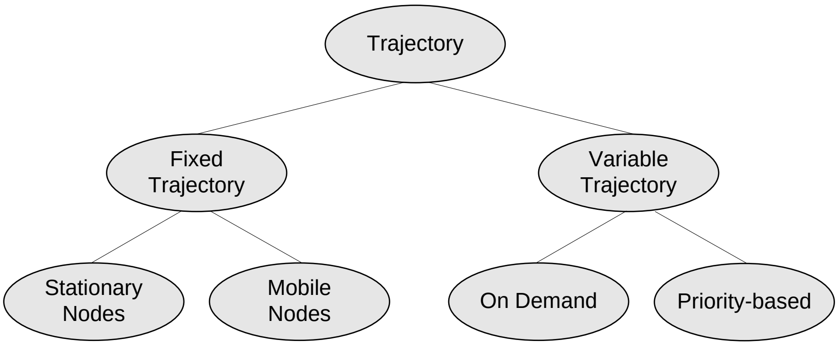

Figure 5. Classification of route control approaches.

5.2. *Speed Control*

The need for speed control comes from the evidence that network connectivity between nodes and MRs may be very different for nodes located at different places and, for each single node, may also vary over time. For example, connectivity may be difficult, or even impossible, due to the presence of physical obstacles. In addition, the wireless link quality may vary significantly from location to location due to the distance between nodes and MRs, the presence of multi-path effects, and so on. For the same location, the wireless link quality may also vary from time to time, e.g., due to meteorological changes. Finally, the throughput experienced by a node depends on the density of nodes in its proximity.

The basic idea behind speed control is to improve the data communication performance by reducing the MR speed in places where the data communication is more difficult (i.e., the net throughput is lower), and increasing it where data communication performs better (i.e., the net throughput is higher). In practice, the speed control module at the MR monitors the throughput experienced in the communication, and adapts the MR speed to improve the data communication performance.

Speed control is also related with power management. As it will be shown in Section 6, from the power management point of view, the time interval between two consecutive visits of the MR to the same node should be fixed or have very small variations. This allows nodes to sleep for the entire time between two successive visits, and save energy. Therefore, the speed control strategy should be designed in such a way to minimize variations in the time between consecutive visits.

Several strategies for speed control have been proposed in the literature. They are summarized below.

- *No Control*. There is no control on the MR speed. This typically occurs when the MR is part of the environment and, hence, its mobility is aimed at providing a service different from message collection/delivery (e.g., a public transportation service). In such a case no control is possible, of course. However, even when the MR is a controllable mobile element (e.g., a mobile robot) dynamic speed control may not be implemented for reducing implementation complexity and/or costs. In such a case the MR moves, for instance, at an approximately constant speed. The range of available speeds is obviously dictated by the mobile element acting as MR. Within this range, the optimal speed value can be chosen by taking into account several parameters, e.g., contact duration of each node, constraints on message latency, energy consumed by the mobile element for locomotion, and so on.

- *Stop and Communicate*. This is the simplest form of speed control. As soon as the MR reaches a node, it stops for the time required by the node to (**i**) transfer all its data to the MR, and (**ii**) to receive messages from the MR, if any. Then the MR moves towards the next node. Without any control on the time spent at different nodes, however, this approach may cause the total time taken for each path traversal to be variable. To avoid this drawback [Kans04] proposes the *Stop to Collect Data* (SCD) algorithm which is targeted to sensor networks. Let T be the maximum time the MR can take to complete a round across the network (T is imposed by constraints on message latency), and let s be the (constant) speed required to cover the entire path in a time less than or equal to T. In the SCD algorithm the MR moves at a constant speed of $2s$ and, thus, it requires a time $T/2$ to traverse the entire path. The remaining $T/2$ interval is used by the MR to stop at nodes to collect data. Specifically, if N is the number of nodes, the MR stops at each node for a time $T/(2N)$. A different distribution of extra-time among network nodes would be possible and, perhaps, beneficial. Though simple, the SCD algorithm allows transferring a greater number of messages per visit with respect to the case without speed control, i.e., when the MR moves at a constant speed s [Kans04].
- *Communication-based Speed Control*. The stop and communicate approach described above does not rely on any data-communication performance index to do speed control. It just stops

for a fixed time when a new node is encountered. In addition, the MR can be either moving at a speed $2s$, or stopped. A finer control can be achieved by learning information about data communication of each single node, and varying the MR speed accordingly, like in the Adaptive Speed Control (ASC) algorithm [Kans04] where the speed is adjusted dynamically depending on the message loss rate experienced in the previous passage. The ASC algorithm is extended in [Soma06] to cope with scenarios where nodes are organized into clusters and transmit their messages to the MR through a cluster-head. In the same paper it is also shown that adding more options to the MR motion actually does not produce any benefit. Again, the ASC algorithm was originally proposed for wireless sensor networks where nodes are assumed to be static. However, it can easily be extended to other scenarios as well.

5.3. *Topology Control*

Topology control is another technique that can be used in combination with or as an alternative of trajectory and speed control. In the context of opportunistic networking based on MRs, the goal of topology control is to dynamically adjust the node's transmitting range so as to achieve the desired contact time with MR while reducing the energy consumed by the wireless interface (which is related to the transmission range). Besides reducing the energy consumption, in dense networks, topology control also reduces the probability of contention when accessing the wireless channel [Sant05].

As shown in Figure 6, given the trajectory and speed of the MR, the contact time duration depends on the node's transmission range. The basic idea of topology control can thus be exploited to derive the level of transmission power that allows the required contact time when the trajectory and speed of the MR are known. In addition, the duration of the contact time could also be adjusted dynamically by varying the node's transmission power and, consequently, its transmission range. This may be useful to cope with variations in the external conditions that affect the communication between the node and the MR (e.g., packet losses due to channel errors, collisions with neighboring nodes, etc.).

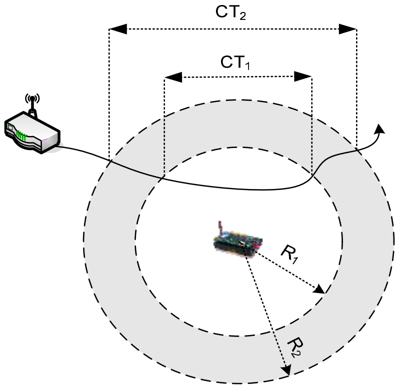

Figure 6. **The contact time can be adjusted by varying the node's transmission range.**

Topology control has been extensively studied in the context of traditional (i.e., multi-hop) ad hoc and sensor networks (a detailed survey can be found in [Sant05]). To the best of our knowledge, only few proposals have been presented in the area of opportunistic networking based on MRs. In [Zhao04] the authors propose a trajectory control technique associated with a sort of topology control based on a dual radio. The MR follows a default trajectory and periodically broadcasts its location using a long-range radio. When a node discovers that the MR is nearby, it sends a service request using its long-range radio. This message contains the node location as well. Upon receipt of a service request, the MR adjusts its trajectory to meet the node. When the two nodes are close enough they start exchanging data using their short-range radio.

6. Power Management and MR Discovery

Since nodes are typically energy-constrained devices, a power management strategy is needed to save energy and increase nodes' lifetime. In the context of opportunistic networking the objective of power management is to minimize energy consumption while missing as few contacts as possible to achieve an adequate performance level in terms of message latency and delivery ratio. Ideally, the node should sleep for most of the time and wakeup only when the MR is within its communication range. In practice, this is unfeasible because the node is not able to know exactly when the next contact will occur, unless the MR mobility pattern is known in advance (predictable mobility). Thus, the MR and the nodes agree on a discovery protocol that allows a timely MR discovery to the node with minimum energy consumption. Obviously, the discovery protocol can be optimized based on the knowledge available about the MR mobility.

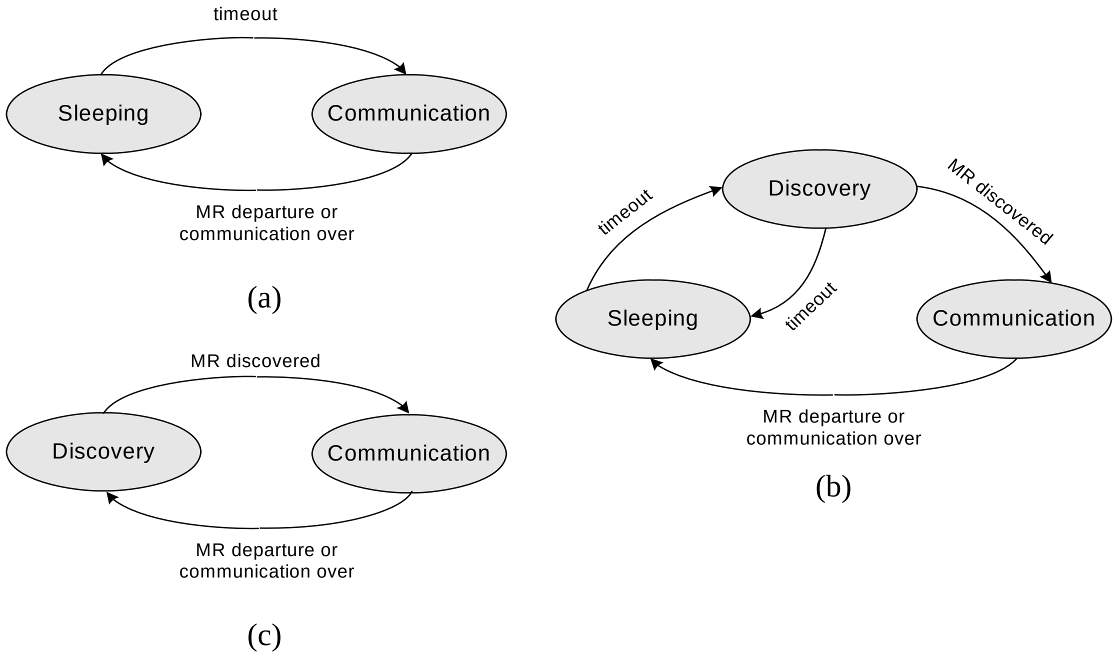

Figure 7. Transitions between different power modes under different degrees of knowledge about MR mobility: complete knowledge (a), partial knowledge (b), and no knowledge (c).

The following scenarios have been identified in [JunA05].

- *Complete Knowledge*. The time between two consecutive contacts with the MR is known in advance by the node (predictable mobility). This may happen when the MR is implemented on top of a controllable mobile element (e.g., a robot), or on a carrier with fixed path and schedule (e.g., a bus shuttle with fixed schedule). Under the assumption of predictable MR mobility, a node can sleep for the time between two consecutive contacts, and wake up only for the time strictly needed to exchange messages with the MR. Obviously, in this scenario the power consumption is minimized [Chak03, JunA05]. Figure 7-a shows a generic node transition between power modes during its activity. Upon departure of the MR (or when there are no more messages to exchange), the node calculates the time to the next contact, sets up a *sleeping timer*, and transitions to the sleeping mode. As soon as the sleeping timer expires, the node wakes up, enters the communication mode, and is ready for exchanging messages with the MR. Again, when the communication is over, or the MR exits the node's communication range, the node returns back to sleep.

- *Partial Knowledge*. In practice, it is not very common to have a complete knowledge of MR mobility. However, even if it is not known in advance, the MR behavior may be learned by observing successive MR passages. By exploiting learning techniques, the node can derive statistics about contact duration and time between contacts (e.g., mean, variance, distribution). Needless to say, the efficiency of power management depends on the degree of knowledge the node has about the MR mobility. Figure 7-b shows the node transition diagram for this specific scenario. Let us assume the node is initially in communication mode. It remains in this mode until there are messages to exchange and the MR is within communication range. Then, the node derives an estimate of the time to the next contact, and sets up a timer accordingly. Upon timer expiration the node enters the discovery mode. Unlike the previous scenario, now there is no guarantee that the MR is within the communication range when the node wakes up. In the discovery mode the node is waiting for the MR arrival. To this end, the node and the MR implement a distributed discovery algorithm to allow a timely MR detection by the node (see below). To reduce energy consumption nodes typically operate on a low duty cycle while in the discovery mode. In addition, the node remains in the discovery mode for a maximum discovery timeout. Then, it assumes that the contact was missed. Hence, it estimates the time to the next contact, sets up the sleeping timer, and switches back to the sleeping mode. Conversely, as soon as the node realizes that the MR has entered its communication range, it switches to the fully operational mode (i.e., 100% duty cycle) and enters the communication mode to perform message exchange.

- *No Knowledge*. The worst scenario is when there is no information available about MR mobility. For instance, this may occur when MRs move randomly through the network, and no assumption can be done about the times the MR will visit a node. In such a scenario, it is impossible to derive statistics (e.g., about the time between contacts). Therefore, each node must remain continuously active looking for possible MR arrivals. The node transition diagram for this scenario is depicted in Figure 7-c. Since the time a node passes in the discovery mode may be very large the discovery algorithm must be very energy efficient, so as to allow a timely discovery of MR while keeping the energy consumption low.

6.1. *Discovery Algorithms*

The discovery algorithm is a distributed algorithm used to allow a node to detect the presence of the MR as it enters the node’s communication range. As the discovery phase may take a long time (especially in the no-knowledge scenario) to be carried out, energy efficiency should be of primary concern in the design of the discovery algorithm. Energy efficiency is typically achieved by putting nodes in a low duty cycle while in the discovery mode. Duty cycle reduces energy consumption but, at the same time, it also increases the *discovery latency*, i.e., the time interval taken by a node to detect the MR presence inside its communication range. Obviously, the discovery latency should be as small as possible compared to the duration of the contact time so as to allow a larger amount of traffic to be exchanged between the node and the MR during the contact duration. The efficiency of a discovery scheme can be measured by means of the discovery ratio defined as the average value of the discovered contact-time (i.e., contact time less discovery latency) divided by the contact time [JunA06], i.e.,

$$\eta = E\left[\frac{contact\,time - discovery\,latency}{contact\,time}\right].$$

The design of a discovery algorithm must reach a tradeoff between energy saving and discovery ratio. Of course, the discovery algorithm can be customized to the specific application scenario. Ideally, the most efficient discovery scheme consists in waking up a node exactly when the MR enters its communication range. This allows the maximum discovery ratio at the minimum energy cost. Unfortunately, this approach is difficult to implement in practice. In fact, its applicability is limited to the predictable mobility scenario where MR visit times are known in advance to each node, and the clocks of nodes and MR are synchronized. For all the other cases, a different approach must be used.

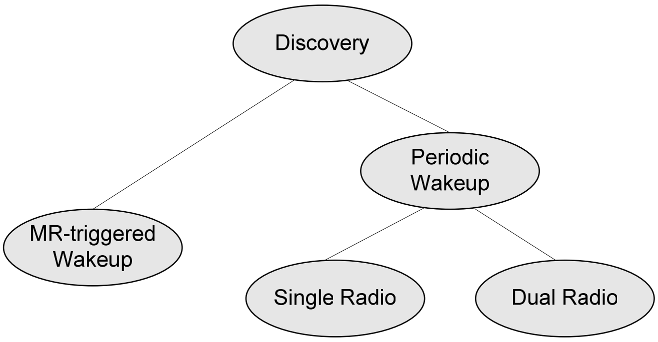

Figure 8. Classification of discovery approaches.

Figure 8 shows the main approaches proposed in the literature for discovery techniques. They can be broadly classified into two categories: *MR-triggered wakeup* and *Periodic wakeup*. In the former scheme nodes are passive (i.e., their radio is kept in sleep mode during the discovery phase), and are awakened by the MR itself when it enters the node’s communication range [GuSt05]. In the periodic wakeup scheme nodes wake up periodically to look for possible MR arrivals [Shah03, Kans04, JunZ05, JunA05, JunA06, Jain06]. Finally, for more efficient energy consumption, MR discovery and data communication can be performed over two different radio channels (when available): a low-power channel for MR discovery, and a high-power channel for data communication [Jain06, JunA06].

7. Relevant Case Studies

In this section we provide some relevant case studies on opportunistic systems based on mobile relays. Section 7.1 describes a scenario that may arise after a disaster when the existing infrastructure is unusable, and airplanes or terrestrial vehicles can be used as ferries to transport data between users in separated areas. Similarly, energy efficient data collection in sensor networks can be performed by using mobile elements that can be either part of the external environment (Section 7.2) or part of the network infrastrucuture (Section 7.3). Finally, Section 7.4 focuses on a special case of opportunistic sensor networks, i.e., underwater sensor networks with MRs.

7.1. *Message Ferrying*

A message ferrying scheme is a scheme to provide connectivity in *sparse* mobile ad hoc networks which are characterized by sparse node deployment and network partitions that may last for extended periods of time. [Zhao04] assumes dealing with this kind of networks and introduces extra mobile nodes named *Message Ferries* to offer a service of message relaying. *Message Ferries* move around in the network area and collect messages from the source nodes then, they provide forwarding of the collected messages. Message collection may happen in two ways:

- *Node Initiated Message Ferrying*: the ferry node moves around following a pre-defined and known path. Each node in the network has knowledge of the paths followed by active ferries. The node wishing to deliver a message moves towards the nearest ferry and when sufficiently close, forwards its messages. Hence, the source node changes its trajectory to meet up with the ferry. This may obviously cause some degradation in the performance achieved by those applications that are currently running on the node during route deviation. Therefore, the source node controls its trajectory towards the ferry while striving to balance between performance degradation in the running tasks and performance gain in data delivery (i.e., minimizing message drops).
- *Ferry-Initiated Message Ferrying*: the ferry node, again, moves around following a pre-defined, default path. To let other regular nodes know its position with good approximation, the ferry periodically sends its position information via long-range broadcast signals. Any source node wishing to deliver messages sends a `ServiceRequest` to the ferry also via a long-range radio signal. The source node also includes its current position in the `ServiceRequest`. After having received the request from the source node, the ferry changes its trajectory to meet up with the source node. The source node periodically communicates `LocationUpdates` to let the ferry adjust its trajectory in order to meet with it. The new trajectory of the ferry is computed with the aim to minimize message drops. When the ferry and the source node are close enough message exchanges occur by means of short-range radio signals.

In both cases each node is expected to have location awareness, i.e., to know its position as well as the position of the ferry/ies, for example through GPS receivers.

A key issue of the message ferrying scheme is the design of the best trajectory that the ferry/ies should follow to service the nodes. The trajectory/ies design goal is to meet the traffic demand while minimizing the delay of data delivery. Obviously, better results can be met when multiple ferries are active in the network area, even though some effort should also be spent for their coordination or even synchronization in some cases. When a single ferry is available in the network area, good results are obtained by designing a route which consists of an ordered sequence of *way-points* and *waiting times* corresponding to these way-points [Tari06]. The ferry node traverses this so-determined route repeatedly and waits for the pre-defined waiting time at each way-point so as it can contact every node in the network with a certain probability. In fact, given for each node the probability to visit a particular place, the number and the places of way-points are determined such that the ferry can meet all the nodes with a given minimum probability. After having decided the particular set of way-points, the minimum path traversing them all is computed and this is established to be the ferry trajectory.

By introducing multiple ferries into the network, the overall system becomes much more fault tolerant because even if a single ferry fails in collecting the data, other ferries can intervene in substitution. Moreover, the system becomes more scalable because a wider geographical area is covered and the traffic load is much more balanced over the entire network deployment. However, the presence of multiple ferries in the network also causes some extra costs, for example to define the best trajectories that ferries should follow, and to assign to each ferry the best subset of nodes to serve. Indeed, sometimes also some degree of synchronization among the ferries could be needed, for example when ferries are supposed to exchange messages with each other. Multiple ferries in the network can traverse the same trajectory starting at different moments and keeping fixed distance in between each other. Another possibility is that message ferries are assigned different trajectories, each one serving a specific subset of nodes, but that may also overlap, resulting in some nodes being visited for data collection more frequently than others. The ferry trajectories are computed in such a way to minimize, on average, the weighted delay between each pairs of nodes. The well-known traveling salesmen problem is exploited for this purpose.

Ferries travelling throughout the network can be completely independent each other such that they do not interact in any way, or otherwise they can exchange messages each other so thus to reduce the messages delivery delay. In case ferries exchange messages each other, they can do it directly when they meet each other, or by exploiting the static nodes they visit during the travel, the same that are data sources of the network, as relay nodes. This way a ferry can download the messages it carries to an intermediate stationary node and another ferry visiting later the same node can upload these messages in order to carry them to the destination or to another intermediate node. As it is shown in [Zhao05], the best performance is achieved when multiple ferries travel through different trajectories and each one of them is assigned its own subset of nodes from which to upload data. Moreover, better performance is experienced when ferries do not interact with each other to exchange the messages they carry. In fact, performing ferry relaying is expensive since synchronization between ferries is necessary. The message ferry scheme scales well with the number of ferries in terms of throughput, delay and resource requirements in both nodes and ferries.

7.2. *Data MULEs*

A data-MULE system [Shah03, Jain06] is very similar to a message ferrying scheme. Data-MULE systems are specifically designed for sparse sensor networks and focus on energy saving. They consist of a three-tier architecture:

- The lower level is occupied by sensor nodes that periodically perform data sampling from and about the surrounding environment.

- The middle level consists of mobile agents named *Mobile Ubiquitous LAN Extentions*, or MULEs for short. MULEs move around in the area covered by sensors to gather their data, which have previously been collected and temporarily stored in local buffers. Data MULEs can be for example people, animals, or vehicles too. They move independently from each other and from sensor positions by following unpredictable trajectories. Whenever they get within reach of a sensor they gather information from it.
- The upper level consists of a set of wired Access Points (APs) and data repositories which receive information from the MULEs. They are connected to a central data warehouse where the data received is synchronised and stored, multiple copies are identified, and acknowledgments are also managed.

Sensor nodes are supposed to be immobile and continuously awake waiting for a MULE to pass by for sending data to it. Sensor-to-MULE transmissions make use of short-range radio signals and hence do not consume too much energy. While moving around, when the MULE eventually passes by any AP deployed in the area, it transmits the collected sensors' data to it. MULEs are assumed to move independently one another, each following a Discrete Random Walk mobility model. No data exchange is assumed to occur among the MULEs and, finally, time synchronization is assumed to be present among sensors and MULEs.

Thank to the short-range radio exchanges, the data MULEs' architecture is a very energy-efficient solution for data gathering in sparse sensor networks if compared to solutions based on the introduction of base stations to cover the entire area to monitor and also to solutions based on the introduction of a high number of sensor nodes to form a dense, entirely connected, sensor network. It also guarantees scalability and flexibility against the network size.

Unfortunately, this solution has a couple of limits, both depending on the randomness of the MULEs' motion. Firstly, the latency for data arrival at the APs is considerable because some time elapses from the sampling instant to the moment the MULE takes the data, and then till the time the MULE actually reaches the AP and delivers the data to it. The second drawback is the fact that sensors have to continuously wait for any MULE to pass and cannot sleep. This leads to energy wastage.

When increasing the area to be monitored the frequency of the visits to the sensors by MULEs naturally decreases, and an increase in the buffer size of the sensors is needed to prevent data loss. The latency experienced by the data monitored increases too. This effect can be alleviated by increasing the number of MULEs. When increasing the area to be monitored, the frequency of the visits of MULEs to the APs decreases too. This leads to a further increase in the latency of data and to the need to increase the buffer size at the MULEs to prevent data loss. An increase in the number of APs can help alleviate the above effects. In conclusion, the number of MULEs can be traded for the size of the sensors' buffers whereas the number of APs can be traded for the size of the MULEs' buffers.

7.3. *Mobile Controllable Infrastructure*

[Soma06] addresses energy-efficient data collection from sparse wireless sensor networks through a mobile infrastructure consisting of a mobile base station. The primary purpose of this approach is to save part of the energy generally spent by sensor nodes in multi-hop transmissions towards a static sink node. In the framework developed the mobile base station moves along a pre-determined path which is fixed. Sensor nodes which are located in proximity of the mobile base station path send their data directly to the base station when in the communication range. Nodes which are far apart from the path followed by the base station send their data over a multi-hop path towards the base station when it passes by or alternatively to one of the nodes which are positioned near to the path of the base station. These nodes act as data repositories until the base station passes and finally collects all the pieces of data stored. Energy saving is addressed in that a large number of nodes is visited by the base station and can thus transmit data over a single hop connection using short range radio. The other nodes which are not in proximity of the path followed by the base station send their data over a multi-hop path which is however shorter, and thus cheaper, with respect to the path established towards a fixed sink node in a classical dense wireless sensor network.

To manage this kind of data collection, nodes self-organize into clusters where cluster heads are the nodes which are nearer to the path of the base station whereas the other nodes of the cluster send their data to the cluster head for storage until the next visit of the base station. Data from the sensor nodes of the cluster travels towards the cluster heads according to the directed diffusion protocol. Election of the cluster heads is done after the first traversal of the base station. During this first traversal the base station does not collect any data.

Transmissions from cluster heads to the base station occur only when the base station is in proximity so as not to waste energy in useless transmissions. Hence, the base station periodically broadcasts POLL messages to inform of its approaching. Cluster heads that receive the POLL message from the base station start sending data to it. The base station acknowledges receipt of each message from a cluster head to inform it that the connection is still active and that the data is reliably delivered. Retransmissions are managed by cluster heads for the messages which are not acked. A cluster head stops transmitting when either it has sent out all the messages stored in cache or it realizes that the connection to the base station is lost for not having received a POLL message for a certain time period.

The trajectory of the base station can be controlled both in space and time. However, changing the trajectory of the base station is not always possible in case of sensor networks because sensors may be deployed in places with obstacles, on rough terrain, or generally where unmanned vehicles can move only in certain directions. Hence, having a fixed path could often be a system requirement rather than a choice. Controlling the trajectory in time instead is considered to be a much more interesting possibility. The base station can move at a constant speed worked out, for example, depending on the buffer constraints of the cluster heads. Each cluster head is thus visited before its buffer runs out of space. However, better performance is experienced when the base station alternates between two states: moving at a certain constant speed or stopping. So base stations move fast in places with no, or only a few, sensors and stop in proximity of cluster heads where sensor deployment is denser. The determination of places where sensor deployment is denser (congested regions) is done at each traversal of the base station. The base station registers the identity of each node it has received a message from and the number of messages received from it. Given that each sensor node collects data at the same rate and thus has the potential to send the same number of packets, the only reason why some nodes send fewer messages than others is that they are in a congested area with more sensors served by the same cluster head which cannot succeed in sending all the data buffered during the limited visit of the base station. In the next traversal the base station stops for more time in regions which have previously been found congested.

7.4. *Underwater Sensor Networks*

Underwater Sensor Networks [Vasi05] are recently attracting lot of attention in the research community. They are deployed to monitor and model the behaviour of the underwater ecosystems. They exploit the aforementioned data MULE communication system. These sensor networks gather physical variables such as water temperature, pressure, conductivity, turbidity, and also pollutants' concentration. Moreover, underwater sensors may collect images to measure visible changes in the deep underwater environment or even to classify species. The network consists of both static nodes and mobile nodes. Static nodes are sensor nodes that perform data collection and storage. They are extremely power efficient because have little energy available and they are not easily rechargeable. Mobile nodes are *Autonomous Underwater Vehicles (AUVs)* which are responsible for data collection from the sensor nodes. They navigate the network to be within communication range of sensors to collect data from them. AUVs require much more power than sensor nodes because they navigate the sensor network however, they are quite easily rechargeable. Static nodes are mostly in deep sleep mode and wake up every few seconds to determine if they are being signalled by mobile nodes nearby. Relieving static nodes from most of the communication and storage loads contributes maximizing the network lifetime.

It has been found that the most efficient way for collecting data from an underwater sensor network is using a system capable of both optical and acoustic communications. Optical communications guarantee high data rate and high bandwidth but need line-of-sight to be established between the communication peers and can only cover short ranges. Acoustic communications on the other hand have the potential for higher transmission range but suffer from attenuation and reflections and allow lower bandwidth. Therefore, a trade-off needs to be met between communication range and data rate. Due to the broadcast nature of acoustic communication, when an acoustic transmission is holding, any other node is prevented to transmit, even to signal an event. Nevertheless, an optical communication and an acoustic one may hold simultaneously. Hence it has been established that the optical communication system is used for short-range line-of-sight data transfers between sensor nodes and AUVs (data mules).

These transmissions are aimed at downloading the stored data from the sensor nodes and uploading commands to them. As they may involve much data, a faster communication system is more appropriate to use. The acoustic system is instead used to signal events over long distances and to transmit small amounts of data. Signalling an event allows the AUV to move to the area of interest, and may also trigger redeployment of the sensor network to concentrate on some important features in the environment. Acoustic communications are indeed particularly suitable for sensor node localization. In fact, the speed of sound in water is low enough to permit accurate timing of signals to determine the distance between nodes. Pair-wise node distances are then used to perform 3D localization. The tasks of the mobile node are to establish a tour of the network, locate each node in the tour, one at a time, and hover above each node to download the data optically. During this period of communication the mobile node may also upload data to the static node, for example to adjust its clock or to change the data sampling rate. The key challenges for underwater data muling are a) locating the first node of the sequence to visit, b) locating the next nodes of the sequence, c) controlling the hover mode (for the mobile node), d) accomplishing data transfer, and e) synchronizing clocks so that the data collected by the sensor network is consistently time stamped. Localization of the first node of the data muling tour starts by positioning the robot in the general area of the network. Given that the general location is known in GPS coordinates, the AUV can perform surface navigation guided by GPS to move toward the node. Once close the AUV descends to the optical communications range. At this point the AUV performs a spiral search to locate the node by making use of distributed localization algorithms built on top of acoustic ranging.

8. Conclusions

In this chapter we have provided a survey of routing approaches to opportunistic networks. This is a very hot topic, since opportunistic networks provide solutions for intrinsically disconnected ad hoc networks, which is one of the main points missing from the research on the legacy MANET paradigm. We have described in detail one of the most interesting cases of opportunistic networks, i.e., the Mobile Relay Forwarding (MRF) approach. MRF assumes that a small subset of nodes have fewer restrictions in terms of resource constraints and follow completely different mobility patterns with respect to the vast majority of nodes in the network. These nodes, called Mobile Relays, can be for example mounted on buses roaming in a city, while regular nodes can be pedestrians' devices. Mobile Relays are opportunistically exploited by the other users to bring messages to the destination, thus connecting nodes that would never be connected together, or anyway significantly improving the network performance. We have discussed a number of issues addressed in the literature with respect to MRF. Specifically, we have provided an extensive taxonomy of the system with respect to the type of Mobile Relay mobility, and we have discussed the case in which the mobility of Mobile Relays can be controlled. We have also described how power-management can be achieved in such a scenario. Finally, we have presented some relevant case studies highlighting how the Mobile Relay Forwarding concept can be exploited also in different scenarios.

Despite the vast body of research in the field, there are still a number of open questions. Just to name a few we can highlight MR motion control techniques, MR discovery under unpredictable (but learnable) mobility patterns, power management and data communication protocols improving the performance of simple *stop-and-wait* protocols.

9. References

[Akka05] K. Akkaya, M. Younis, and M. Bangad, “Sink Repositioning for Enhanced Performance in Wireless Sensor Networks”, *Computer Networks*, Vol. 49 (2005), pp. 512-534.

[Anas07] G. Anastasi, M. Conti, E. Monaldi, and A. Passarella, “An Adaptive Data-transfer Protocol for Sensor Networks with Data Mules”, *Proc. IEEE International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM 2007)*, Helsinki (Finland), June 2007.

[Bold07a] C. Boldrini, M. Conti, I. Iacopini, and A. Passarella, “HiBOp: a History Based Routing Protocol for Opportunistic Networks”, *Proc. IEEE WoWMoM 2007*, Helsinki, Finland, June 2007.

[Bold07b] C. Boldrini, M. Conti, and A. Passarella. “Impact of Social Mobility on Routing Protocols for Opportunistic Networks”, *Proc. IEEE WoWMoM AOC Workshop 2007*, Helsinki, Finland, June 2007.

[Burn05] B. Burns, O. Brock, and B. N. Levine, “MV Routing and capacity building in disruption tolerant networks”, *in Proceedings of the IEEE INFOCOM 2005*, Miami, FL, March, 2005.

[Chak03] A. Chakrabarti, A. Sabharwal, and B. Aazhang, “Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks”, *Lecture Notes in Computer Science 2634, Proc. International Workshop on Information Processing in Sensor Networks (IPSN 2003),* Springer, 2003.

[Gand03] S. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, “Energy-Efficient Schemes for Wireless Sensor Networks with Multiple Mobile Base Stations”, *Proc. IEEE Globecom 2003, S. Francisco (USA)*, December 1-5, 2003.

[Good97] D. Goodman, J. Borras, N. Mandayam, and R. Yates, “INFOSTATIONS: A new system model for data and messaging services”, *Proc. IEEE VTC’97*, Vol. 2, May 1997, pp. 969–973.

[GuBo05] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C. Lee, “Partitioning Based Mobile Element Scheduling in Wireless Sensor Networks”, *Proc. IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2005),* pp. 386- 395, S. Clara, USA, Sept.2005.

[GuBo06] Y. Gu, D. Bozdag, and E. Ekici, “Mobile Element Based Differentiated Message Delivery in Wireless Sensor Networks”, *Proc. IEEE Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM 2006)*, Nyagara Falls (USA), June 2006, pp. 83-92.

[GuSt05] L. Gu and J. Stankovic, “Radio-Triggered Wake-up for Wireless Sensor Networks”, *Real-Time Systems Journal*, Vol. 29, pp. 157-182, 2005.

[Harr06] K. Harras and K. Almeroth, “Inter-Regional Messengers Scheduling in Delay Tolerant Mobile Networks”, *Proc. IEEE International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM 2006)*, Nyagara Falls (USA), June 2006, pp. 93-102.

[Jain06] S. Jain, R. Shah, W. Brunette, G. Borriello, and S. Roy, “Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks”, *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 11, no. 3, June 2006, pp. 327-339.

[JeaS05] D. Jea, A. Somasundra, and M. Srivastava, “Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks”, *Proc. IEEE Int’l Conference on Distributed Computing in Sensor Systems (DCOSS* 2005), Marina del Rey, USA, 2005.

[Jetc03] J. G. Jetcheva, Y.-C. Hu, S. PalChaudhuri, A. K. Saha, and D. B. Johnson, “Design and Evaluation of a Metropolitan Area Multitier Ad Hoc Network Archtecture, *Proc. of IEEE WMCSA 2003,* Monterey, CA, October 2003

[Juan02] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, “Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with Zebranet,” *Proc. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2002.

[JunA05] H. Jun, M. Ammar, and E. Zegura, “Power Management in Delay Tolerant Networks: A Framework and Knowledge-Based Mechanisms”, *Proc. IEEE Conference on Sensor and Ad Hoc Communication and Networks* (SeCon 2005), September 2005.

[JunA06] H. Jun, M. Ammar, M. Corner, and E. Zegura, “Hierarchical Power Management in Disruption Tolerant Networks with Traffic-aware Optimization”, *Proc. ACM Workshop on Challenged Networks (CHANTS 2007)*, Pisa (Italy), September 2006.

[JunZ05] H. Jun, W. Zhao, M. Ammar, E. Zegura, and C. Lee, ”Trading Latency for Energy in Wireless Ad Hoc Networks using Message Ferrying”, *Proc. IEEE PerCom Workshops, International Workshop on Pervasice Wireless Networking* (PWN 2005), March 2005.

[Kans04] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, and D. Estrin, “Intelligent Fluid Infrastructure for Embedded Networks,” *Proc. Second Int’l Conf. Mobile Systems, Applications, and Services (MobiSys)*, 2004.

[Legu06] J. Leguay, T. Friedman, and V. Conan, “Evaluating Mobility Pattern Space Routing for DTNs”, *in Proceedings of the IEEE Infocom 2006,* Barcelona, Spain, April 2006.

[Lind03] A. Lindgren, A. Doria, and O. Schelèn, “Probabilistic routing in intermittently connected networks”, ACM Mobile Computing and Communications Review, 7(3), July 2003.

[LuoH05] J. Luo and J.-P. Hubaux, “Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks”, *Proc. IEEE INFOCOM 2005,* Miami (USA), March 13-17, 2005.

[Muso05] M. Musolesi, S. Hailes, and C. Mascolo, “Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks”, in Proceedings of the 6th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2005), Taormina-Giardini Naxos, Italy, June 13-16, 2005.

[Pelu06a] L. Pelusi, A. Passarella, and M. Conti, “Encoding for Efficient Data Distribution in Multi-hop Ad hoc Networks”, *Handbook of Wireless Ad hoc and Sensor Networks*, A. Boukerche, Editor, Wiley and Sons Publisher (to appear).

[Pelu06b] L. Pelusi, A. Passarella, and M. Conti, “Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad hoc Networks”, *IEEE Communication Magazine*, Vol. 44, N. 11, November 2006, pp. 134-141.

[Sant05] P.Santi, “Topology Control in Wireless Ad Hoc and Sensor Networks”, ACM Comp. Surveys, Vol. 37, n. 2, p. 164-194, June 2005.

[ShaR03] R. C. Shah, S. Roy, S. Jain, and W. Brunette, “Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks, *Proc. IEEE Int’l Workshop on Sensor Network Protocols and Applications (SNPA 2003),* May 11, 2003, pp. 30- 41.

[Smal03] T. Small and Z. Haas, “The Shared Wireless Infostation Model — A New Ad Hoc Networking Paradigm (or Where There Is a Whale, There Is a Way),” *Proc. ACM MobiHoc, 2003.*

[Soma04] A. Somasundara, A Ramamoorthy, and M. Srivastava, “Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines”, *Proc. IEEE International Real-Time Systems Symposium (RTTS 2004)*, Lisbon (Portugal), December 5-8, 2004.

[Soma06] A. Somasundara, A. Kansal, D. Jea, D. Estrin, and M. Srivastava, “Controllably Mobile Infrastructure for Low Energy Embedded Networks”, *IEEE Transactions on Mobile Computing*, Vol. 5, N. 8, August 2006.

[Sush04] J. Sushant, K. Fall, and R. Patra, “Routing in a delay tolerant network,” in *Proceedings of SIGCOMM’04*, August, 2004.

[Tari06] M. M. B. Tariq, M. Ammar, and E. Zeruga, “Message Ferry Route Design for Sparse Ad hoc Networks with Mobile Modes”, *Proc. of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2006),* Florence, Italy, May 22-25, 2006.

[Vahd00]	A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks”, *Tech. Rep. CS-2000-06, Department of Computer Science, Duke University*, Durham, NC, 2000.
[Vasi05]	I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin and P. Corke, “Data Collection, Storage, and Retrieval with an Underwater Sensor Network”, *Proc. of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, San Diego, November 2-4, 2005.
[Wang05]	Z. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, “Exploiting Sink Mobility for Maximizing Sensor Network Lifetime”, *Proc. Hawaii International Conference on System Science (HICSS-38)*, Big Island (USA), January 2005.
[Widm05]	J. Widmer and J.-Y. Le Boudec, “Network Coding for Efficient Communication in Extreme Networks”, *Proc. of ACM SIGCOMM 2005 Workshop on delay tolerant networks*, Philadelphia, PA, USA, August 22–26, 2005.
[Zhao03]	W. Zhao and M. Ammar, “Message Ferrying: Proactive Routing in Highly-Partitioned Wireless Ad Hoc Networks,” *Proc. of IEEE Workshop on Future Trends in Distributed Computing Systems (FTDCS 2003)*, Puerto Rico (USA), May 2003.
[Zhao04]	W. Zhao, M. Ammar, and E. Zegura, “A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks”, *Proc. of ACM Mobihoc 2004*, Tokyo (Japan), May 2004.
[Zhao05]	W. Zhao, M. Ammar, and E. Zegura, “Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network”, *Proc. of IEEE INFOCOM 2005*, Miami (USA), 2005.