Marco Fotino

Evaluation and Optimization of the Energy Behavior of Routing Protocols for Mobile Ad-hoc Networks

ING-INF/03

– Monograph –

November 29, 2010

Springer
Berlin Heidelberg New York
Hong Kong London
Milan Paris Tokyo

Abstract

MANETs (Mobile Ad-hoc NETworks) are networks made up entirely of wireless devices, without the support of any fixed infrastructure. Because of their auto-configuring and self-managing features, and of their many application fields, MANETs have been one of the most investigated research fields in recent years.

In particular, the interest of the scientific community is aimed at routing protocols for such networks, which should solve the problem of efficient multi-hop routing in a distributed environment.

My research work is focused on an aspect of MANETs that in recent years has become increasingly important: the evaluation of routing algorithms in terms of energy consumption. My work for the Systems Engineering and Computer Science Ph.D. program is focused on the study of routing algorithms for mobile ad-hoc networks and performance evaluation of such networks, especially trying to highlight and resolve issues related to energy consumption of mobile devices.

The study of mobile ad-hoc networks and of the routing algorithms has been carried out mainly through simulations. The algorithms and metrics for routing protocols, obtained by the adaptation, the improvement and the joint application of the solutions known in literature, have been implemented in the ns-2 network simulator software. The methodology used in the context of the Ph.D. program consists in the implementation of the algorithms and in the execution of a large number of simulations, in order to validate the effectiveness of the adopted solutions in the widest possible number of scenarios.

Keywords – *MANET, Routing, Energy, OLSR, GPSR*

Contents

1

Introduction

In recent years, thanks to the proliferation of wireless devices, the use of mobile networks is growing rapidly. In particular, a very large number of recent studies have focused on Mobile Ad-hoc Networks (MANETs, [1]).

A MANET is a network without a fixed infrastructure, in which every node can act as a router (Figure 1.1); this is required when the two end-points interchanging data are not directly within their radio range. This kind of network, self-organizing and self-reconfiguring, allows a huge range of applications, from rapidly auto-configuring networks (for example within a conference or a business meeting) to hard environments (like a battlefield or a natural disaster scenario, Figure 1.2). The ideal applications include settings that deal only with mobile nodes, including home networks, search and rescue operations, vehicular networks. Furthermore, new applications are rapidly developing for MANETs.

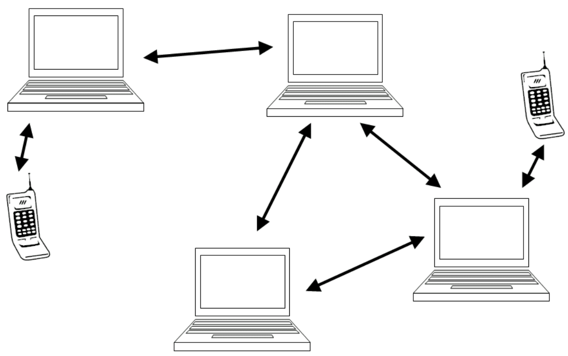

Fig. 1.1. Example of Mobile Ad-hoc Network

Fig. 1.2. Rescue operations in a disaster scenario

Performance of a mobile ad hoc network heavily depends on the selected routing scheme, and traditional Internet routing protocols do not work efficiently in a MANET. This kind of network, in fact, has a dynamic topology (every node can move randomly and the radio propagation conditions change rapidly over time) and a limited bandwidth (so that the control traffic overhead must be reduced to the minimum). However, not all routing protocols developed perform well in a given situation: factors such as mobility, network size, network load, bandwidth and signal strength do affect the performance of MANET routing protocols.

The design of MANET routing protocols is a challenging task and it has been an extensive research area in recent years. Many protocols have been proposed, from a variety of perspectives. These protocols try to satisfy various properties, like: distributed implementation, efficient bandwidth utilization, throughput optimization, fast route convergence and freedom from loops. The purpose of routing protocols is to establish the shortest, correct and most efficient route between a pair of nodes. The routing protocols for MANETs can be distinguished into three main categories: proactive, reactive, and hybrid protocols.

The proactive protocols exchange control messages between nodes periodically to maintain a consistent view of the network even when there is no active data session. This allows the proactive protocol to discover the route quickly at the price of large bandwidth consumption from the overhead in exchanging control message. Moreover, there is a waste of network resources because every node has maintain the complete view of the network even though most routing information is never used.

In contrast, the reactive protocols establish and maintain the route between the source and the destination only if there is a request. Because of this feature, these protocols are also called on-demand protocols. The established route is maintained as long as the data session is active. After a certain period of time, when the data session becomes inactive, the route is removed to release the occupied resources. Therefore, reactive protocols consume less bandwidth than proactive protocols. However, due to the dynamic characteristic of the ad hoc network, the packet might suffer a variable and long delay as the route discovery/recovery might have to be performed at each hop it travels through. To overcome these weaknesses of both proactive and reactive protocols, the hybrid protocols have been proposed.

In hybrid protocols, groups of nodes (often called zones), are formed and a proactive routing method is used within each zone while a reactive routing method is used to communicate with remote nodes. Most hybrid protocols separate nodes into flat zones and a few use hierarchical structures like trees or clusters. With this method, the overhead is reduced because the inefficient control overhead of the proactive approach is limited only within the zone and the lower overhead of the reactive protocol is used to efficiently connect each zone. However, the performance of hybrid protocols often relies on the trade-off in parameters like the zone radius, which needs to be particularly adjusted to each network before use.

More details of existing MANET routing protocols and their features are discussed in Chapter 2.

1.1 Research Objective

Since mobile hosts today are powered by battery, efficient utilization of battery energy is a key factor. Battery life, therefore, can also affect the overall network communication performance: when a node exhausts its available energy, it ceases to function and the lack of mobile hosts can result in partitioning of the network. For that reason, reducing power consumption is an important issue in ad hoc wireless networks. Only a few of the routing proposals to date have focused on the power constraints of wireless nodes: traditional routing protocols tend to use the shortest path algorithms (minimum hop count) without any consideration of energy consumption, often resulting in rapid energy exhaustion for the small subset of nodes in the networks that experience heavy traffic loads.

The problem of energy efficiency in mobile networks must be faced at every network layer: at the physical layer, low consuming devices (CPU, disk, antenna) can be used; at the Medium Access Control layer, transmission power can be adapted to the actual network needs and the interface can be turned off when the node is inactive; at the routing layer, the energy status of nodes can be taken into account when selecting paths; and at the application layer, data can be compressed before transmission to reduce the network load.

For instance, reducing the overhearing effect (i.e. wireless interfaces consuming energy to receive data addressed to other nodes) can lead to a great improvement in overall energy consumption of the network, as demonstrated in [2]. These approaches can, however, affect other classical metrics such as the end-to-end delay and throughput; energy-efficient solutions must take this issue into account, and attempt to reduce every possible side-effect.

The aim of this research work is the evaluation of routing algorithms in terms of energy consumption, focusing on the study of routing algorithms for mobile ad-hoc networks and performance evaluation of such networks, especially trying to highlight and resolve issues related to energy consumption of mobile devices.

The study of mobile ad-hoc networks and of the routing algorithms is carried out mainly through simulations. The algorithms and metrics for routing protocols, obtained by the adaptation, the improvement and the joint application of the solutions known in literature, are implemented in the ns-2 network simulator software. The methodology used in the context of the Ph.D. program consists in the implementation of the algorithms and in the execution of a large number of simulations, in order to validate the effectiveness of the adopted solutions in the widest possible number of scenarios.

1.2 Thesis Structure

The rest of this thesis is organized as follows. First, some of the most important routing protocols for MANETs are presented in Chapter 2. Then, the OLSR protocol, that represents the basis for the research study, is illustrated in details in Chapter 3, while Chapter 4 describes the main energy issues at the routing layer of the ad-hoc networks. In Chapter 5 the analytical and computational instruments used to carry on the study are presented, and in Chapter 6 the energy-aware proposals are described. Finally, the simulation results validating the proposals are illustrated in Chapter 7 and some conclusions and future works are drawn in Chapter 8.

Part I

Research Context

2

Routing Protocols for MANETs

In Chapter 1, the motivation for the impact of routing protocols on MANETs was outlined. This chapter provides a review of literature on MANET routing protocols. Routing protocols are mechanisms to transfer information in data packets from a source to a destination in a network. Generally, two activities take place in routing protocols which enables communication to occur between two nodes. First, some control information is spread over the network, in order to create a shared knowledge of the network topology. Second, different metrics are used to evaluate optimal paths that data packets should use when sending packets in a network. The design of routing algorithms requires careful consideration to achieve its goals in an efficient way.

This chapter is divided into several sections. Section 2.1 outlines the routing protocol design issues. The properties of routing protocols are presented in Section 2.2 and routing approaches to MANETs in Section 2.3. Finally, the most important parameters for the comparison of routing protocols are discussed in Section 2.4.

2.1 Routing Protocols Design Issues

There are many routing protocol issues to consider when designing MANET routing protocols. The design of a routing protocol is very challenging because of the need of a distributed state across a network made of unreliable devices, of the dynamic topology resulting from mobility of nodes, of the limited network capacity in terms of bandwidth and of various types of wireless communication constraints. Some of these constrains are: variable link quality, energy constrained nodes, interference and hidden/exposed terminals.

2.1.1 Distributed state in unreliable environment

In Mobile Ad-hoc Networks, every node can act as a router for its neighbors. That feature enables the multi-hop communications between distant nodes, but makes the communication sensible to any problem at the intermediary nodes. The distribution of resources in any unreliable environment becomes a challenge to enable communication, therefore routing protocols should consider best utilisation of resources like bandwidth, processing power and battery life, in order to avoid the partitioning of the entire network.

2.1.2 Dynamic topology (Mobility)

The network topology in a MANET changes dynamically due to the mobility of nodes, therefore causing sessions of transferring packets to suffer from interference, leading to frequent path breaks. The interference occurs when an intermediate or destination node in a route disappears from the network range. When a path breaks it is important that a routing protocol efficiently seeks to learn new available paths and builds a new topology so that reliable connections are established. Mobility management is extremely important and it justifies the need for efficiency in any MANET routing protocol.

2.1.3 Limited network capacity (Bandwidth)

Unlike wired networks with a large bandwidth, MANETs are limited by the radio channel. Therefore data transfer rates are lower than wired networks ones. For that reason, a routing protocol needs an optimal use of the bandwidth. Furthermore, limited bandwidth permits to store less topology information. A complete topology information is required for an efficient routing protocol, however this cannot be the case in MANET routing protocol as this would cause an increase in node control messages. An efficient routing protocol is required for a balanced usage of the limited bandwidth.

2.1.4 Resource constraints

The two resources which are essential to nodes in MANETs are processing power and battery life. Increasing processing power consumes more battery life. Nodes in a MANET are portable devices, hence processing power and battery life are limited. It is important to design a routing protocol that efficiently allow transfer within the limited life span of battery life using less processing power.

2.1.5 Interference and collisions (Hidden and exposed terminal problems)

During simultaneous transmission of two nodes, collision occurs when each node does not know about each others transmission. The exposed terminal problem contributes to the inability of a node that has been blocked due to transmission of a nearby node to another node, thus the radio reusability spectrum is affected. Transmission cannot occur when the spectrum is in use, so it is important to promote handshakes between neighbor nodes during the communications. On the other hand, the hidden terminal problem (Figure 2.1) occurs when two nodes try to transmit data to the same node, because they are not whitin the carrier sensing range of each other. This causes loss of packets due to collisions, with the consequence of retransmissions and loss of bandwidth. The IEEE 802.11 MAC layer ([3]) use the RTS/CTS mechanism to avoid this problem.

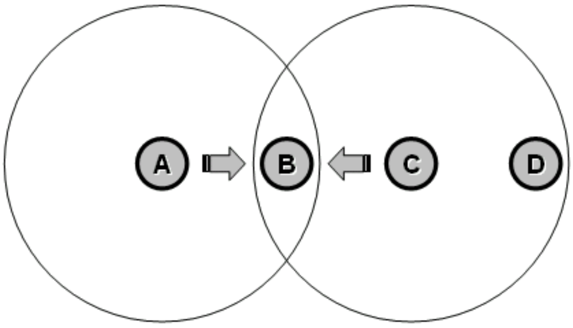

Fig. 2.1. Hidden terminal problem

2.2 Properties of Routing Protocols

The performance of a routing protocol in terms of high throughput, low overhead and limited delay, is determined by the properties of a routing protocol. Routing protocols should be distributed in operation and not to be dependent on a centralised node as centralised operations are not scalable. Since nodes can enter or leave a network due to their mobility, a distributed routing operation is more fault tolerant than a centralised routing operation.

Routing protocols should guarantee that routes supplied are free from loop and are free from stale routes that consume bandwidth and processing power. The efficient routing computation and maintenance is another property required to involve a minimum number of nodes. These nodes are required to have access to the route as quickly as possible within a minimum setup time.

Power conservation is a desirable property as nodes like Laptops and Personal Digital Assistants (PDA) have very limited resources. Therefore an optimal use of resources like bandwidth, processing power, memory and battery life is vital.

Collisions of packets may occur when packets are transferred from source to destination hence a minimum packet collision is a required property. This strategy minimises the collision as much as possible during broadcast messages. This amounts to a reliable reduction of data loss and prevention of stale route occurrences.

The radio environment may cause the presence of unidirectional links. Utilizing unidirectional links can improve routing protocol performance. MANET encounters different types of data packets, some of which may require some Quality of Service (QoS) control. Providence to a certain level of QoS is an essential requirement by some sensitive and real-time applications.

2.3 Routing Approaches for MANETs

Routing protocols for mobile ad-hoc networks have different features regarding the way they exchange information and establish communication. The protocols developed in recent years are classified into three broad categories. These are: proactive, reactive and hybrid routing protocols.

On demand (reactive) routing protocols determine routes only when a node has a data packet to send. A node with a packet to send is referred to as source node. If the route to the destination is not known, the source node initiates a search (route discovery) to find possible routes to the destination ([2]). The optimised route is then used and maintained, establishing connection and communication until such a route is no longer required or becomes invalid. The DSR, AODV and TORA are examples of on demand routing protocols.

Table driven (proactive) routing protocols attempt to maintain consistent and up to date information of all possible routes, to all destinations, at all times, regardless of whether the routes are needed. To support this consistency, the protocol sends flooding messages (broadcast) to spread update information and all possible connectivity through the network ([2]). Proactive protocols require each node to maintain more than one table to store routing information regardless of the need for such route information ([4]). They also share common features, like background information exchange regardless of the communication request strategy employed ([5]). Examples of table driven routing protocols are: Fisheye State Routing (FSR), OLSR, Destination Sequenced Distance Vector (DSDV) and Topology Broadcast Based on Reverse Path Forwarding (TBRPF).

Figure 2.2 outlines the classification of proactive and reactive routing protocols.

2.3.1 Reactive Approach

Reactive routing protocols, also known as on-demand routing protocols, have been proposed to reduce the number of control overhead by maintaining only the information for active routes. Instead of maintaining all the routes at all times, the protocol starts route discovery on-demand. In the route discovery process, a route request packet (RREQ) is usually flooded until it reaches the

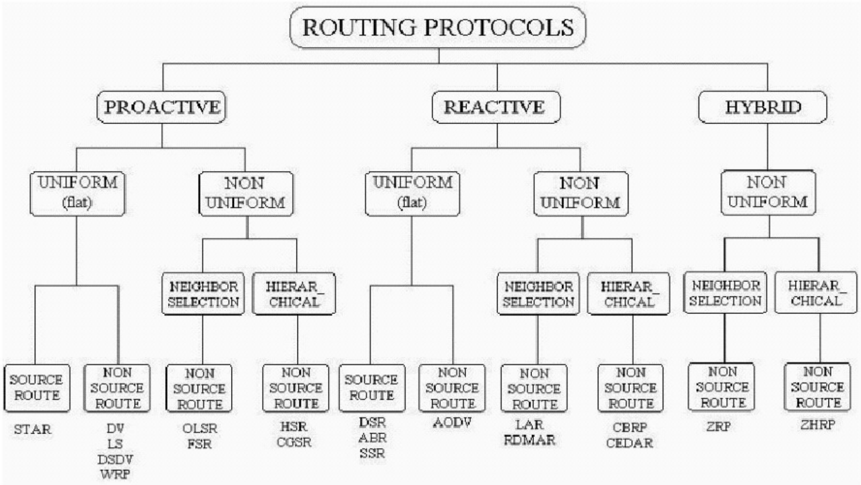

Fig. 2.2. Classification of proactive and reactive routing protocols

destination (or a node that contains the route to the destination). Then, a route reply packet (RREP) is generated and sent back to the source to inform the available route. This route is maintained as long as the connection is active and removed once it is no longer required.

In general, on-demand routing protocols can be classified into 2 categories: hop-by-hop routing and source routing.

Hop-by-hop routing protocols maintain the routing information locally at each node. The data packet stores only the destination address in its header and each intermediate node will use its routing table to forward the packet to the specified destination. The advantage of this approach is the high adaptability of the path because each node can react to the changes in the network faster than the end-to-end manner. However, maintaining the routing information at each node requires higher routing overhead and resources.

Source routing protocols maintain the routing information only at the source. A list of addresses that the packet will traverse until it reaches the destination is embedded into the header of each packet by the source. Each intermediate node has no knowledge of the route to destination and only forwards each data packet by the information in its header. As maintaining the route at each intermediate node is no longer required, the overhead is reduced. However, the probability of route failures could be high when the path becomes long in large networks or when there is a high level of mobility. Moreover, the overhead of embedded route information in the header also affects the performance in large networks. According to these disadvantages, it can be clearly seen that source routing protocols do not scale well.

AODV

An example of hop-by-hop routing protocols is the ad hoc on-demand distance vector routing (AODV, [6]). The standard AODV uses the broadcast route request packet to discover the route to the destination. Once the route request arrives at the destination, the route reply packet is sent back to the source using the reverse route previously established by the route request packet. AODV uses a blacklist to avoid using unidirectional links, which are the links established by the route request packet but cannot be utilized by the route reply packet. Moreover, the precursor list is maintained to keep track of the upstream node that is utilizing this route. When a route failure occurs, a route error packet is sent out in broadcast manner if the precursor list is not empty. This route error packet is repeatedly flooded until it reaches the source node or the node with an empty precursor list. Once the source node receives the route error packet, the route recovery which is the same process as the route establishment using route request and route reply packets is repeated. Regarding the route recovery process, the local route repair feature of AODV can be chosen. Instead of re-initiating the route discovery from the source, the delay can be reduced by initiating it from the node that detects the error. Also, another feature of AODV which allows the intermediate nodes to respond to the RREQ can be chosen to further shorten the delay.

DSR

The Dynamic Source Routing protocol (DSR) is a reactive protocol ([7]). This generates less overhead and provides more reliable routing than proactive routing, but at the cost of finding the optimal route. Mobile hosts do not utilize periodic messages, with a consequent energetic advantage in battery consumption. DSR updates automatically only when it needs to react to changes in the routes currently in use. This protocol is simple and efficient.

DSR uses a modified version of source routing. Operation of the protocol can be divided into two functions route discovery and route maintenance ([7]). Route discovery operation is used when routes to unknown hosts are required. Route maintenance operation is used to monitor the correctness of established routes and to initiate route discovery if a route fails. When a node needs to send a packet to a destination it does not know about, the node will initiate route discovery. The node sends a route discovery request to its neighbors (Figure 2.3). Neighbors can either send a reply to the initiator or forward the route request message to their neighbors after having added their address to the request message (i.e., source routing) as shown in Figure 2.4.

The route reply message can be returned to the initiator in two ways. If the host that sends reply already has the route to the initiator, it can use that route to send the reply. If not, it can use the route in the route request message to send the reply. The first case is beneficial in situations where a

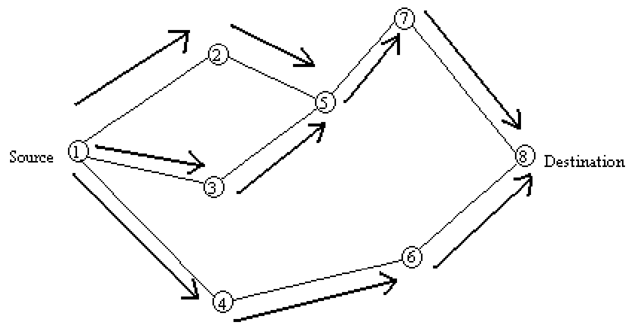

Fig. 2.3. Propagation of Route Request (RREQ) packet

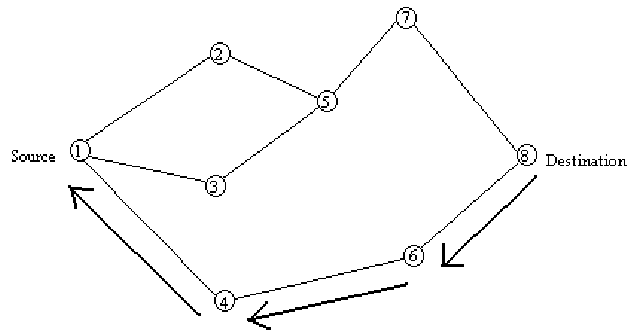

Fig. 2.4. Path taken by the Route Reply (RREP) packet

network might be using unidirectional links, and it might not be possible to send the reply using the same route that the route request message took.

The route cache reply mechanism allows an intermediate node to provide to the source the path towards the destination if it is known (Figure 2.5).

Route maintenance is performed when there is an error with an active route. When a node that is part of some route detects that it cannot send packets to the next hop, it will create a Route Error message (RERR) and send it to the initiator of data packets. The RERR message contains the addresses of the node that sent the packet and of the next hop that is unreachable. When the RERR message reaches the initiator, the initiator removes all routes from

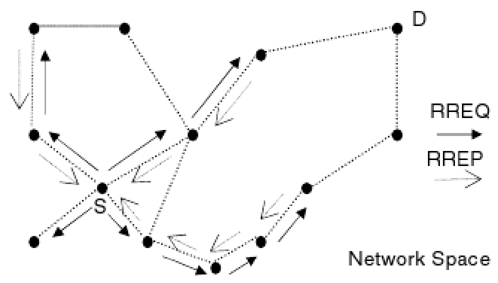

Fig. 2.5. Route Cache (rc) reply

its route cache that have the erroneous node address. It then initiates route discovery for a new route if needed.

The advantages of the DSR protocol include easily guaranteed loop-free routing and very rapid recovery when routes in the network change. The DSR protocol is designed mainly for mobile ad hoc networks of up to about 200 nodes, and is designed to work well with even very high rates of mobility.

ABR

Another source routing protocols is the associativity-based routing for ad-hoc mobile networks (ABR, [8]). ABR is a special case of source routing protocols because it uses a similar route discovery to DSR but also maintain local route information like AODV. Rather than having multiple backup routes like DSR, ABR focuses on the stability of the route. ABR selects the route based on a metric, called associativity tick, which reflects the degree of association stability of mobile nodes. The associativity ticks are maintained by periodic beacons from each node. During the route discovery, not only the addresses are embedded in the packets header but also the associativity tick is included to allow an intermediate node and the destination to select the best path according to all associativity ticks of upstream nodes. As ABR does not have backup routes, a route reconstruction is required upon link failures. Even though this route reconstruction is performed locally, it can still cause a longer delay and more control overhead.

2.3.2 Proactive Approach

In proactive routing protocols, each node attempts to maintain the routing information to every other node by periodically exchanging control messages.

There are many proactive routing protocols and various methods to maintain the routing information. However, they can be classified into 2 categories: distance vector and link state.

Distance vector routing protocols select the path based on the relayed link cost from every other node in the network. In this kind of protocols, every node advertises its directly connected links and their costs along with the relayed link information and costs received from other nodes.

Link state routing protocols maintain a complete view of the network and construct a routing tree for data packet forwarding. To obtain a complete network view, a large amount of routing information is exchanged among nodes. Similar to distance vector protocols, link state protocols also have a high overhead where a large amount of bandwidth is consumed by routing control packets.

DSDV

One example of distance vector routing protocol is the destination-sequenced distance-vector protocol (DSDV, [9]), which uses the number of hops to the destination as the cost. The routing information is advertised in a broadcast manner throughout the network along with the sequence number, which is originally generated by the destination. The sequence number is used to avoid a routing loop problem which is a common problem in distance vector routing. DSDV reacts to the topology changes using two kinds of update packets: full dump and incremental. The full dump packets will carry all available routing information at the current node to another while the incremental packets will carry only the information changed since the last full dump. These two types of routing update packets are used to lower the overhead and shorten the update latency. However, the overhead of DSDV is still large due to the large amount of periodic update information, which makes DSDV not scalable.

WRP

Another example of distance vector protocol is the wireless routing protocol (WRP, [10]). In WRP, each node maintains four tables: a distance table, a routing table, a link-cost table, and a message retransmission list. WRP uses the predecessor information along with the sequence number to avoid routing loops. In addition to the bandwidth consumption overhead, WRP also has a high memory consumption overhead due to the large amount of information maintained at each node.

FSR

One example of link state routing protocols is the fisheye state routing (FSR, [11]). FSR maintains a topology map at each node by exchanging the link state information between neighbor nodes. However, the link state packets are not broadcasted and only periodically exchanged with the local neighbor nodes. FSR reduces the amount of control overhead by removing the event-based link state update and using only the periodic update. Moreover, the periodic update frequency is reduced by the fisheye technique where the node within the smaller scopes updates more frequently than the node that is farther away (Figure 2.6). FSR is based on the global state routing (GSR), which can be viewed as a special case of FSR where the scope is infinite.

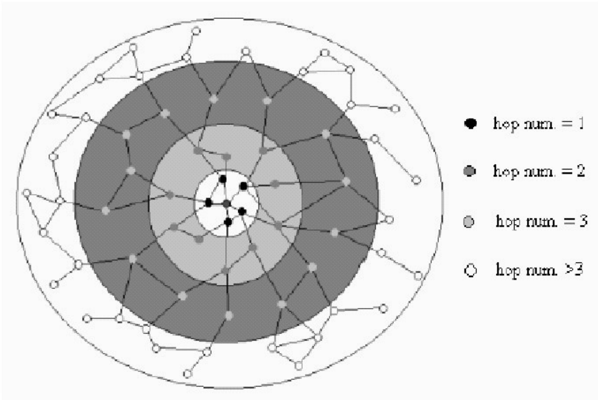

Fig. 2.6. Fisheye state routing

The advantages of FSR are that the flooding is minimized and the routing is more accurate for nodes closer to the destination, which makes it suitable for dense networks. However, the slower update for remote nodes affects the accuracy and by using this imperfect topology information an inaccurate route selection could possibly occur.

OLSR

The Optimized Link State Routing (OLSR, [12]) is a proactive protocol, that represents an optimization of the classical link state algorithm, adapted to the requirements of a MANET. The key concept used in OLSR is that of multipoint relays (MPRs). In order to reduce the effect of flooding messages to all nodes in the network, OLSR selects a subset of nodes, called Multipoint Relays (MPR), to be part of a relaying backbone. In order to build this structure, each node gathers 2-hops neighborhood information and elects the smallest number of relays so that all 2 hops neighbors are covered by at least one relay. Nodes notify the respective relays of their decision so that each relay maintains a list of nodes, called Multipoint Relaying Selectors (MPR Selectors), which have elected it as MPR. Finally, the relaying decision is made on the basis of last-hop address according to the following rule.

Definition 2.1. *(MPR flooding) A node retransmits a packet only once after having received the packet the first time from an MPR selector.*

Figure 2.7 illustrates the MPR mechanism by means of an example: a node can reach its 2-hop neighborhood through a number of connections, but it selects only 3 of its 7 neighbors as MPRs. This way, every 2-hop neighbor of the node is still reachable, and can receive information sent by the central node. It can be demonstrated that the MPR mechanism still finds the shortest paths between every source-destination in the network, although it needs less retransmissions of broadcast information over the network than a classical link state algorithm. In the example, only the black nodes will forward control information sent by the central node; as a consequence, all the traffic addressed to that node will be sent to one of its MPRs.

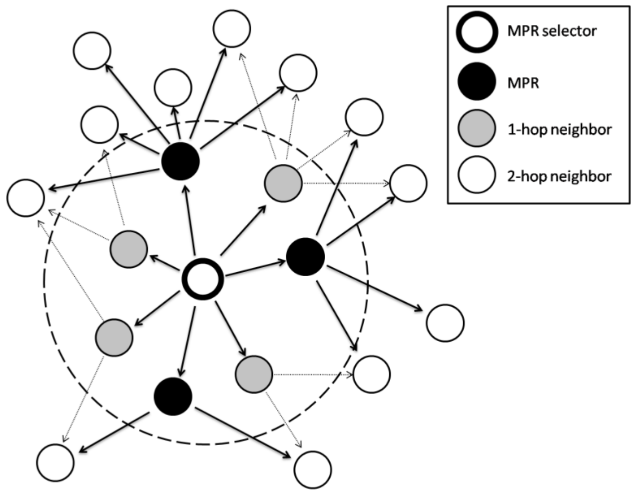

Fig. 2.7. MultiPoint Relay Mechanism in OLSR Protocol

Since OLSR is the reference protocol for the work developed in this thesis, its functionalities will be described in details in Chapter 3.

2.3.3 Hybrid Approach

Hybrid routing protocols use the combination of proactive routing and reactive routing concepts for the purpose of increasing scalability. In hybrid protocols, the network is partitioned into zones. A proactive routing method is used within each zone while a reactive routing method is used to communicate with nodes that are outside of the zone. With this method, the overhead is reduced because the inefficient control overhead of the proactive approach is limited only within the zone and the lower overhead from reactive routing is used to efficiently connect each zone.

ZRP

The first example o hybrid routing protocol is the zone routing protocol (ZRP, [13]). ZRP reduces the proactive routing overhead by limiting the scope of the routing information within a zone (Figure 2.8). A zone is defined by the hop distance between nodes, where the nodes within ρ hops from the current node are in the same zone. ZRP discovers the path to the node outside the zone using bordercasting, which also reduces the number of flooding messages. In bordercasting, the route request packet is forwarded only by the border node of the current zone. When the route request packet is received, the border node looks up the proactive routing table in its zone and sends back the route reply packet if it has the route to the destination, otherwise it repeats the bordercasting process. The routing zone radius ρ is a very crucial parameter in ZRP which also becomes a disadvantage for the protocol. The radius ρ must be carefully chosen based on the network features. If the radius is too large, then ZRP behaves more like a pure proactive protocol. On the other hand, if the radius is too small, then ZRP behaves more like a pure reactive protocol. In both cases, ZRP loses its advantage of reduced overhead and scalability.

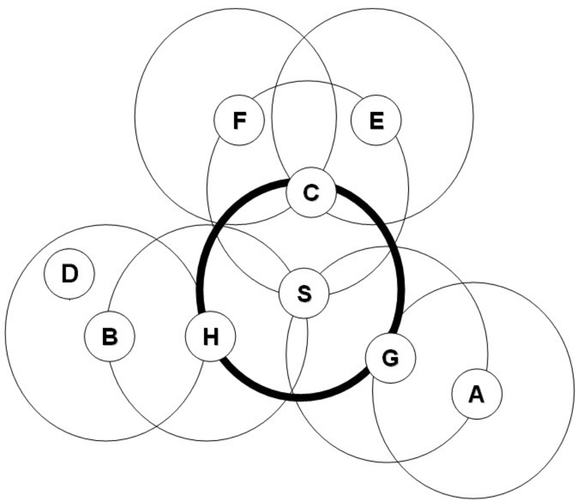

Fig. 2.8. Example of Routing Zone in ZRP

ZHLS

Another example of hybrid routing protocol is the zone-based hierarchical link state (ZHLS, [14]). ZHLS is a zone-based hierarchical link state routing protocol, which also uses the location information from the global positioning system (GPS). In contrast to ZRP which defines overlapping zones, ZHLR utilizes the location information to partition the network into non-overlapping zones and assign each node a node ID and a zone ID. The hierarchical topology consists of two levels: a node level and a zone level. There is no cluster-head in ZHLS because the zones are pre-designed with regard to their location information. Hence, a single point of failure or bottlenecks can be avoided in ZHLS despite being a hierarchical routing protocol. The routing mechanism consists of intra-zone proactive routing and inter-zone reactive routing which is similar to ZRP. Therefore, similar advantages can be achieved. Additional advantages of ZHLS are the fixed zone location. Once the source node knows the node ID and the zone ID of the destination, even if the link breaks, ZHLS can still easily find another route to the destination with less overhead compared to reactive routing protocols. However, this fixed zone location is also the disadvantage of ZHLS as it is required to be preprogrammed before use.

2.3.4 Location-based Protocols

In the last few years, thanks to the proliferation of mobile devices with positioning hardware (like GPS antennas), the routing protocols that base their decisions on geographical information are assuming an increasing importance for the researchers. The efficient utilization of location information, in facts, can make a MANET routing protocol more scalable and can lead at the same time to an improved performance in termo sof overhead.

GPSR

The Greedy Perimeter Stateless Routing (GPSR, [15]) protocol is a geographical protocol, laying on the hypothesis that each node in the network knows its geographical position (for example, reading its own coordinates from a GPS device). It consists of two routing methods: greedy forwarding, used as long as possible, and perimeter forwarding, used only when the former method can not be applied.

The routing process is performed ina per-hop basis: every node having to forward a packet to a destination tries to send it to the neighbor that minimizes the geographical distance to the destination (Figure 2.9).

In this algorithm each node needs only to know the current position of its neighborhood (resulting in a very high robustness to topological changes) and the location of the destination.

In some cases the greedy forwarding fails: the path to destination requires the packet to cover a longer distance to actually reach the destination (as shown in Figure 2.10). GPSR solves this problem (known as “void region problem”) using the perimeter forwarding mechanism.

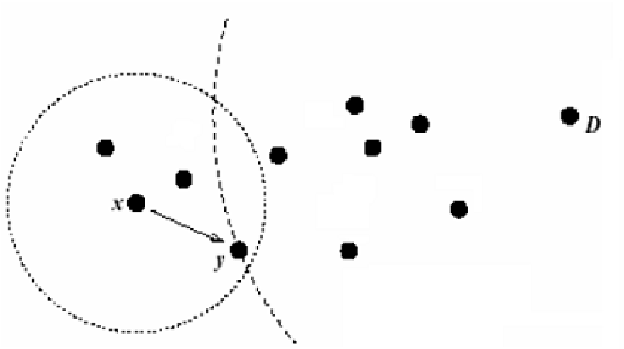

Fig. 2.9. Example of Greedy forwarding in GPSR

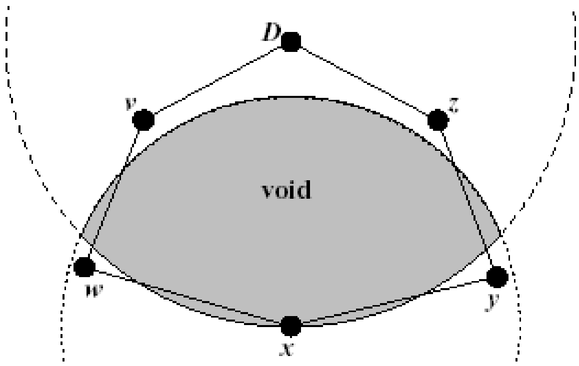

Fig. 2.10. Example of Perimeter forwarding in GPSR

2.4 Comparison of Routing Protocols

The task of routing protocols is to establish and enable transfer of data packets from a source to a destination node in a MANET. Each routing protocol acts differently to enable connections and maintain route. This section provides a comparison for the routing properties outlined in Section 2.3.

2.4.1 Proactive protocols features

Advantages

- paths are immediately available (low delay)
- every node has an updated view of network topology

Disadvantages

- high bandwidth and power consumption for control information update
- slow convergence (especially distance vector protocols)
- big size of routing tables
- low scalability

2.4.2 Reactive protocols features

Advantages

- paths are computed only as needed
- no periodic updates needed (power and bandwidth saving)

Disadvantages

- higher average delay for packet delivery
- the packet header size is higher for longer paths
- the caching of paths at the intermediate nodes requires memory
- sensitive to node mobility

2.4.3 Performance parameters

The main performance paramenters used in literature for the evaluation of routing protocols for MANETs are described in the following.

Throughput

Throughput represents the measurement of the quantity of data transmitted in the unit of time. One of the main targets ofrouting protocol optimization is to reach an higher throughput without increasing the end-to-end delay. It can be obtained by the use of optimal paths or exploiting the cooperation among neighboring nodes.

Average packet delay

In multi-hop enviroments, a packet travelling from a source to a destination is forwarded by a certain number of nodes before it reaches the destination. The total delay of a packet is given by the sum of all the delays on every link that forwarded the packet. At every intermediate node, the delay is composed by the elaboration delay (the time between the reception of the whole packet and its placement in a forward queue), the queue delay (the time spent by the packet in the transmission queue), the transmission delay (the time between the transmission of the first and the transmission of the last bit of the packet) and the propagation delay (the time between the transmission of the last bit of the packet and its reception at the destination). The following equation summarizes the total end-to-end delay:

$$d_{total} = d_{elab} + d_{queue} + d_{tx} + d_{propag} \tag{2.1}$$

A routing algorithm should mantain a constant delay in high traffic conditions, and deliver the packets with the lowest possible delay in normal conditions. It can be noticed how optimal paths in terms of number of hops help to maintain a low end-to-end delay. On the other hand, trying to lower the delay with higher transmission ranges can result in a diminuition of the throughput, as the increased noise introduced affects the bandwidth.

Optimal paths

Routing algorithms should select the best path between every couple of nodes, based on a given metric. When more than one path is available between two nodes, the protocol selects the one that minimizes a cost function. The definition of such cost function is a key factor in the design of a routing protocol.

Convergence

In a mobile environment, paths are subjected to frequent changes, caused by the loss of a link. The routing algorithms react to those events updating their information and recalculating the paths according to the new topology. A good algorithm ensures a rapid convergence to a new set of optimal paths in response to every topology change. A slow convergence, on the other hand, often leads to routing loops and to the loss of data packets. ono causare loop o persino disconnessioni nella rete.

Overhead

The overhead of a protocol consists of all the packets transmitted for the functioning of the protocol itself. It includes, generally, all the routing updates, the control information, and the acknowledgments needed by the routing algorithm. Since the bandwidth of a wireless network is a lmited resource, it is important to have the minimum possible protocol overhead, so that the bandwidth can be used mainly for data transmission.

3

The OLSR Protocol

The Optimized Link State Routing Protocol (OLSR) is designed for mobile ad hoc networks. The protocol is documented in the experimental Request For Comment (RFC) 3626 ([12]). OLSR is table-driven and pro-active and utilizes an optimization called Multipoint Relaying for control traffic flooding. RFC3626 modularizes OLSR into core functionality, which is always required for the protocol to operate, and a set of auxiliary functions. The core functionality specifies a protocol able to provide routing in a stand-alone MANET. Each auxiliary function provides additional functionality, which may be applicable in specific scenarios, e.g., in case a node is providing connectivity between the MANET and another routing domain.

3.1 Node addressing

OLSR uses an IP address as the unique identifier of nodes in the network. As OLSR is designed to be able to operate on nodes using multiple communication interfaces, every node must choose one IP address that is set to be its main address.

OLSR can be used both with IP version 4(IPv4) and version 6(IPv6). In an OLSR context the differences between IPv4 and IPv6 is the size of the IP addresses transmitted in control messages, the minimum size of messages and the address to use as destination for control traffic.

3.2 Information repositories

As a derivate of the classical link state algorithm, OLSR maintains state by keeping a variety of databases of information. These information repositories are updated upon processing received control messages and the information stored is used when generating such messages. Here follows a brief look at the different information repositories used in core OLSR.

Multiple Interface Association Information Base

This dataset contains information about nodes using more than one communication interface. All interface addresses of such nodes are stored here.

Link Set

This repository is maintained to calculate the state of links to neighbors. This is the only database that operates on non-main-addresses as it works on specific interface-to-interface links.

Neighbor Set

All registered one-hop neighbors are recorded here. The data is dynamically updated based on infor- mation in the link set. Both symmetric and asymmetric neighbors are registered.

2-hop Neighbor Set

All nodes, not including the local node, that can be reached via an one-hop neighbor is registered here. Notice that the two hop neighbor set can contain nodes registered in the neighbor set as well.

MPR Set

All MPRs selected by the local node is registered in this repository. The MPR concept is explained in Section 3.4.

MPR Selector Set

All neighbors that have selected this node as a MPR are recorded in this repository.

Topology Information Base

This repository contains information of all link-state information received from nodes in the OLSR routing domain.

Duplicate set

This database contains information about recently processed and forwarded messages.

3.3 Control traffic

All OLSR control traffic is to be transmitted over UDP on port 698. This port is assigned to OLSR by the Internet Assigned Numbers Authority(IANA). The RFC states that this traffic is to be broadcasted when using IPv4, but no broadcast address is specified. When using IPv6 broadcast addresses does not exist, so even though it is not specified in the RFC, it is implicit understood that one must use a multicast address in this case.

3.3.1 Packet format

All OLSR traffic is sent in OLSR packets. These packets consist of a OLSR packet header and a body as displayed in Figure 3.1.

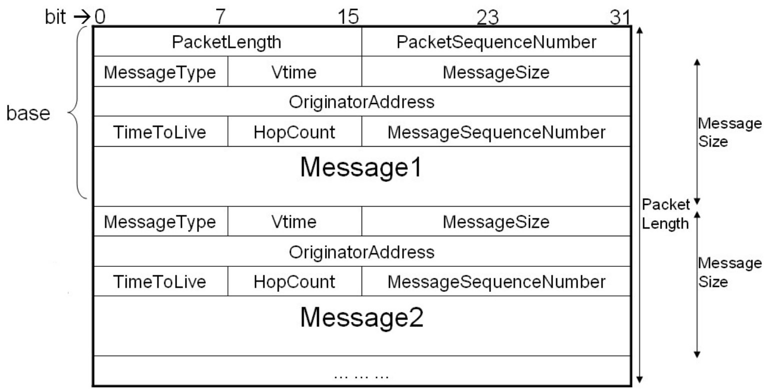

Fig. 3.1. The generic OLSR packet

The fields in the OLSR packet header are:

- Packet Length - The length in bytes of the entire packet, including the header
- Packet Sequence Number - A sequence number incremented by one each time a new OLSR message is transmitted by this host. A separate Packet Sequence Number is maintained for each interface so that packets transmitted over an interface are sequentially enumerated.

An OLSR packet body consists of one or more OLSR messages. OLSR messages use a header as shown in Figure 3.1. All OLSR messages must respect this header. The fields in the header are:

- Message type - An integer identifying the type of this message. Message types of 0-127 are reserved by OLSR while the 128-255 space is considered private and can be used for custom extensions of the protocol

- Vtime - This field indicates for how long after reception a node will consider the information con- tained in the message as valid. The time interval is represented in a mantissa-exponent format
- Message Size - The size of this message, including message header, counted in bytes
- Originator Address - Main address of the originator of this message
- Time To Live - The maximum number of hops this message can be forwarded. Using this field one can control the radius of flooding
- Hop Count - The number of times the message has been forwarded
- Message Sequence Number - A sequence number incremented by one each time a new OLSR packet is transmitted by this host

3.3.2 Message types

The core functionality of OLSR defines tree message types, which will all be described in detail later. All core functionality of OLSR is based on the processing and generation of these messages.

However, the OLSR protocol packet format allows for a wide variety of custom packets to be transmitted and flooded to the needs of the designer. OLSR will forward unknown packet types according to the default forwarding rule as explained later. The MPR optimization used in OLSR makes this possibility for message flooding a great asset to anyone in need of net-wide broadcasting of traffic in the ad-hoc network.

3.4 MultiPoint Relaying

OLSR uses flooding of packets to diffuse topology information throughout the network. Flooding, in its simplest form, means that all nodes retransmits received packets. To avoid loops, a sequence number is usually carried in such packets. This sequence number is registered by receiving nodes to assure that a packet is only retransmitted once. If a node receives a packet with a sequence number lower or equal to the last registered retransmitted packet from the sender, the packet is not retransmitted. On wired networks other optimizations are usually added such as no retransmission on the interface on which a packet arrived.

On a wireless multi-hop network however, it is essential that nodes retransmits packets on the same interface that it arrived, since this is the very nature of wireless multi-hop networks. This again causes every re-transmitter to actually receive a duplicate packet from every symmetric neighbor that re-transmits the packet. A wireless flooding scenario is depicted in Figure 3.2. One can see that every transmission leads to a reception of the same packet. The originator of the flood could be any node in the figure.

The number of retransmissions using traditional flooding is $n-1$ where n is the number of nodes in the network. In our case (Figure 3.2) it will be 24. This flooding technique can clearly benefit from some sort of optimization.

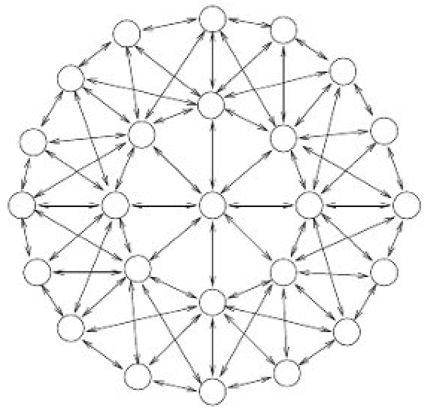

Fig. 3.2. Flooding a packet in a wireless multihop network. The arrows show all transmissions

3.4.1 MultiPoint Relays

The concept of multipoint relaying is to reduce the number of duplicate retransmissions while forwarding a broadcast packet. This technique restricts the set of nodes retransmitting a packet from all nodes, to a subset of nodes. The size of this subset depends on the topology of the network.

This is achieved by selecting neighbors as Multipoint relays(MPRs). Every node calculates its own set of MPRs as a subset of its symmetric neighbor nodes chosen so that all 2-hop neighbors can be reached through a MPR. This means that for every node n in the network that can be reached from the local node by at minimum two symmetric hops, there must exist a MPR m so that n has a symmetric link to m and m is a symmetric neighbor of the local node. In the scenario illustrated in Figure 3.3, node selects the black nodes as MPRs. This way all two hop nodes can be reached through a MPR. Node will not retransmit traffic from that is to be flooded.

OLSR lets nodes announce their own willingness to act as MPRs for neighbors. Eight levels of willingness are defined from the lowest `WILL_NEVER` (0), which indicates that this node must never be chosen as a MPR, to the highest `WILL_ALWAYS` (7), which indicates that this node should always be chosen as a MPR. The willingness is spread through `HELLO` messages and this information must be considered when calculating MPRs.

Finding the optimal MPR set has been proved to be a NP-complete problem. RFC 3626 proposes a rather simple heuristic for MPR calculation.

3.4.2 Forwarding OLSR traffic

Relaying of messages is what makes flooding in MANETS possible. OLSR specifies a default forwarding algorithm that uses the MPR information to

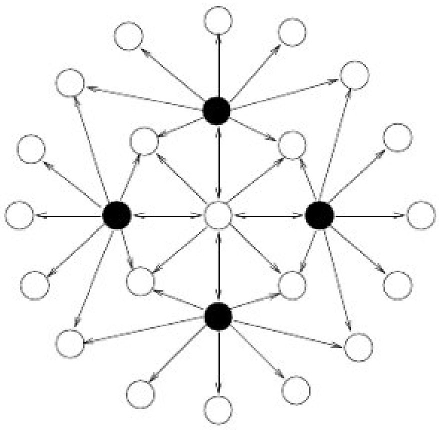

Fig. 3.3. Flooding a packet in a wireless multi-hop network from the center node using MPRs(black). The arrows show all transmissions

flood packets. One is however free to make ones own rules for custom forwarding of custom messages. But all messages received that carries a type not known by the local node, must be forwarded according to the default forwarding algorithm. The algorithm can be outlined as follows:

1. If the link on which the message arrived is not considered symmetric, the message is silently dis carded. To check the link status the link set is queried.
2. If the TTL carried in the message header is 0, the message is silently discarded.
3. If this message has already been forwarded the message is discarded. To check for already forwarded messages the duplicate set is queried.
4. If the last hop sender of the message, not necessarily the originator, has chosen this node as a MPR, then the message is forwarded. If not, the message is discarded. To check this the MPR selector set is queried.
5. If the message is to be forwarded, the TTL of the message is reduced by one and the hop-count of the message is increased by one before broadcasting the message on all interfaces.

The fact that all received unknown message types are forwarded using this approach makes flooding of special message-types possible even if these message-types are only known to a subset of the nodes.

The number of retransmissions in a MPR scenario highly depends on the network topology and the MPR calculation algorithm. Using the same topology as in Figure 3.2, a possible MPR calculation could lead to the black nodes in Figure 3.3 being chosen as MPRs by the center node. As one can see, if the center node is to flood a message throughout the network, 4 retransmissions are done using MPR as opposed to 24 using traditional flooding.

The duplicate set

To be able to check if a message has already been retransmitted, a cache of recently processed and forwarded messages is maintained. The data stored is the minimum needed to identify the message. This means that the actual message content is not stored, but rather just originator address, message-type and sequence number. This data is cached for a constant time of `DUP_HOLD_TIME`, suggested to be 30 seconds in the RFC. Every received message that is processed by the local node is registered in the duplicate set. If the message is forwarded, the duplicate-entry representing this message is updated accordingly, registering on what interfaces the message has been forwarded. Based on querying the duplicate set, a node can then keep track of already processed messages and already forwarded messages on a per-interface basis.

Forward jitter

To avoid radio collisions due to synchronized forwarding, a jitter is introduced to the message forwarding. This is a random small time interval for which the message is to be cached in the node before forwarding it. When using forwarding-jitter, piggybacking of messages will often occur since multiple messages that are to be forwarded might arrive within the buffer period. When this happens, messages are stacked within the same OLSR packet.

3.4.3 Link set optimization

Due to the nature of the MPR selection, only nodes which are chosen as MPRs by one or more neighbors, needs to declare their link state. In facts, these nodes need only to declare the MPR selectors in the link state messages. When this information is flooded to all nodes in the MANET, all nodes will have enough information to calculate shortest path routes to all hosts. The default OLSR setting is that a node only floods link state messages if it is chosen as MPR by at least one neighbor, and it only announces its MPR selectors in these messages. Only the nodes selected as MPRs by one or more neighbors will transmit link-state messages. One can easily see that this information, in addition to some neighbor-sensing scheme, will be sufficient to create a full understanding of the topology.

3.5 Neighbor discovery

Obviously, OLSR needs some mechanism to detect neighbors and the state of the communication lines to them. `HELLO` messages are emitted on a regular interval for this purpose. A very simplified version of a neighbor discovery session using `HELLO` messages, is displayed in Figure 3.4. `A` first sends an empty `HELLO` message. `B` receives this message and registers `A` as an asymmetric neighbor due to the fact that `B` can not find its own address in the `HELLO` message. `B` then sends a `HELLO` declaring `A` as an asymmetric neighbor. When `A` receives this message it finds its own address in it and therefore sets `B` as a symmetric neighbor. This time `A` includes `B` in the `HELLO` it sends, and `B` registers `A` as a symmetric neighbor upon reception of the `HELLO` message.

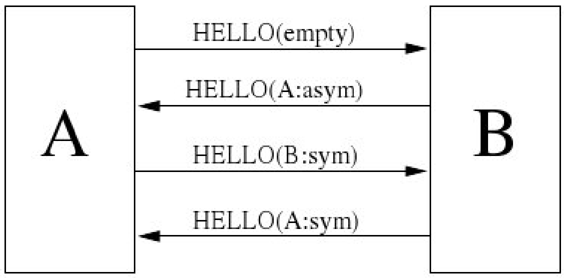

Fig. 3.4. A typical neighbor discovery session using HELLO messages

But `HELLO` messages serves other purposes as well. They are generated and transmitted to all one-hop neighbors to achieve link-sensing, neighbor-sensing, two-hop neighbor-sensing and MPR selector sensing.

In `HELLO` messages nodes transmit information about all known links and neighbors. The types of the neighbors are also declared. This includes declaring what MPRs the node has selected. Registered links and neighbors are grouped by the link and neighbor type to optimize byte usage. It is very important to note that `HELLO` messages are generated on a per-interface basis. This is because `HELLO` messages are used for link sensing, which requires the use of possible non-main-addresses.

The format of the `HELLO` message can be seen in Figure 3.5. This message is included as the body part of an OLSR-message in an OLSR packet as seen in Figure 3.1. The 8 byte link-code contains both information about the link to the neighbor and the type of the neighbor. The link type describes the state of the link and the neighbor type describes the state of the neighbor including MPR information. Note that a link can be set as asymmetric while the neighbor is still set as symmetric, if multiple links to the neighbor exist.

3.5.1 Link sensing

To keep up-to-date information on which links exist between a node and its neighbors, the link set is maintained. In `HELLO` messages a node emits all

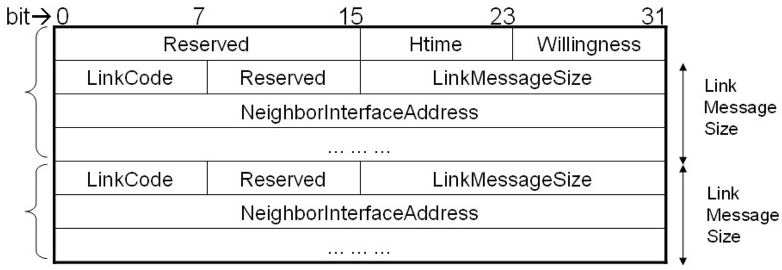

Fig. 3.5. The OLSR HELLO message

information about the links to neighbors from the interface on which the `HELLO` is transmitted. When declaring links, the IP addresses of the actual interfaces making up the link is used. When declaring the neighbor state of neighbors not reachable on the interface on which the `HELLO` is transmitted, the main address of the neighbor node is used.

Upon receiving a `HELLO` from a neighbor, a node checks to see if the `HELLO` message contains the IP address of the interface the message was received. The link set is then updated as follows:

- If no link entry exists for the tuple (originating IP, IP of received interface) then such an entry is created. The originating IP is fetched from the IP header of the received packet. Whenever a link entry is created a corresponding neighbor entry is created as well if no such entry exists.
- An asymmetric timer is then updated according to the validity time received. This timer decides for how long the link entry is to be considered asymmetric if the symmetric timer times out.
- If the address of the receiving interface is located in the received `HELLO` message, the symmetric timer is updated and the status of the link is updated if necessary. The status of the neighbor entry according to this link entry is also updated if necessary.
- Finally the actual holding time for this entry is set to be the maximum of the asymmetric timer and the symmetric timer.

3.5.2 Neighbor detection

Neighbor detection populates the 1-hop neighbor repository and only uses the main addresses of nodes. As seen in the previous section, the neighbor entries are closely related to the link entries. Whenever a link entry is created, the neighbor table is queried for a corresponding neighbor entry. Note that this neighbor entry must be registered on the main address of the node. If no such entry can be located, then a new neighbor entry is created. This means that while a node can have several link-entries describing different links to the same neighbor, only one neighbor entry exists per neighbor.

The status of the neighbor entries is also updated according to changes in the link-set. A neighbor is said to be a symmetric neighbor if there exists at least one link-entry in the link set connecting a local interface to one of the neighbors interfaces where the symmetric timer is not timed out. When a link-entry is deleted, the corresponding neighbor entry is also removed if no other link entries exist for this neighbor.

3.5.3 Two-hop neighbor detection

A node also maintains a repository of all nodes reachable via symmetric neighbors. This is the two hop neighbor set. This database is used for MPR calculation.

Upon receiving a `HELLO` message from a symmetric neighbor, all reported symmetric neighbors, not including addresses belonging to the local node, are added or updated in the two hop neighbor set. Entries in the two hop neighbor set are all based on main addresses, so for all received entries in the `HELLO` message the MID (Multiple Interface Declaration) set is queried for the main address. Note that the two hop neighbors also may contain neighbors reachable by one hop.

3.5.4 MPR Selector detection

The MPR flooding scheme is based on the requirement that nodes have registered which neighbors have chosen them as a MPR. Nodes mark their selected MPR neighbors in `HELLO` messages by setting the Neighbor Type to be `MPR_NEIGH`.

Upon receiving a `HELLO` message, a node checks the announced neighbors in the message for entries matching one of the addresses used by the local node. If an entry has a matching address and the neighbor type of that entry is set to `MPR_NEIGH`, then an entry is updated or created in the MPR selector set using the main address of the sender of the `HELLO` message.

3.6 Link state declaration

Link state routing protocols are based on nodes flooding the network with information about their local links.OLSR uses host based flat routing, so the link state emitted describes links to neighbor nodes. This is done using Topology Control(`TC`) messages. The format of a `TC` message is shown in Figure 3.6.

`TC` messages are flooded using the MPR optimization. This is done on a regular interval, but `TC` messages are also generated immediately when changes are detected in the MPR selector set. In OLSR the flooding process itself is optimized by the usage of MPRs, but as explained in Section 3.4.3, the MPR technique introduces two link-state declaration optimizations as well. OLSR

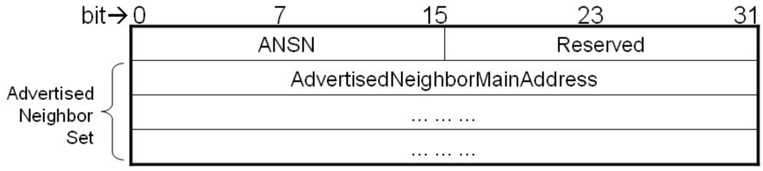

Fig. 3.6. The OLSR Topology Control message format

nodes can also be tuned to send more than just its MPR selector set. One should notice that more robust routing could be achieved by announcing more than the MPR selector set.

The MPR functionality introduces two optimizations to `TC` messaging:

Size optimization

The size of `TC` messages is reduced due to the fact that a node may only declare its MPR selectors in `TC` messages. The factor of this reduction is related to how dense the network topology is. In a topology as shown in Figure 3.3 the `TC` message size of the center node would be reduced to half the size of a "classical" `TC` message(not including headers). When using IPv6, a simple example like this reduces a net-wide broadcast message of 64 bytes.

Sender optimization

Nodes that has no links to declare usually does not transmit `TC` messages. The exception here is nodes that just lost their MPR selectors. These nodes are to generate empty `TC` messages for a given interval to update the nodes in the MANET.

But except from this special case, if only declaring MPR selectors in `TC` messages, only nodes selected as MPRs will generate `TC` messages. Such a reduction in actual transmitted messages greatly reduces the overall overhead of control traffic.

3.6.1 Advertised Neighbor Sequence Number

The Advertised Neighbor Sequence Number (ANSN) is a sequence number associated with a nodes advertised neighbor set. However, this number is not increased on every `TC` generation. The ANSN represents the "freshness" of the information contained in the message. This means that whenever a node detects a change in its advertised neighbor set the ANSN is increased. Keep in mind that the advertised neighbor set in a node can, as described later, vary from only the MPR selectors to the entire symmetric neighborhood.

3.6.2 Populating the topology set

Upon receiving a TC message, the TC repository is updated as follows:

- If no entry is registered in the TC repository on the address of the originator, one is created with validity time and ANSN set according to the TC message header.
- If an entry is registered in the TC repository on the address of the originator and with ANSN lower than the received ANSN, then that entry is updated according to the received TC message.
- If an entry is registered in the TC repository on the address of the originator with an ANSN equal to the received ANSN, then the validity time of the entry is updated.

3.7 Route calculation

The proposed heuristic for route calculation in RFC3626 is a relatively trivial shortest-path algorithm. It can be outlined as:

1. Add all 1-hop neighbors registered as symmetric to the routing table with a hop-count of 1.
2. For each symmetric one-hop neighbor, add all two hop neighbors registered on that neighbor that has:
   - not already been added to the routing table
   - a symmetric link to the neighbor

   These entries are added with a hop-count of two and next-hop as the current neighbor.
3. Then, for every added node N in the routing table with hop-count $n = 2$ add all entries from the TC set where:
   - the originator in the TC entry $== N$
   - the destination has not already been added to the routing table

   New entries are added with a hop-count of $n + 1$ and next-hop as the next-hop registered on N's routing entry.
4. Increment n and do step 3 over until there are no entries in the routing-table with hop-count $== n + 1$

4

Energy Issues in MANETs

Since mobile hosts today are powered by battery, an efficient utilization of battery energy is a key factor for this kind of networks. Moreover, battery life can also affect the overall network communication performance: since every node in a MANET acts as a router, when a node exhausts its available energy, it ceases to function and the lack of mobile hosts can result in partitioning of the network, because of unreachable destinations. For that reason, reducing power consumption is an important issue in ad hoc wireless networks.

The limited capacity of the batteries of mobile nodes introduces a strong constraint for MANETs. In the last few years, many routing protocols have been developed for MANETs, but often with little consideration of energy issues. Only recently researchers have focused on finding strategies to reduce energy consumption and to prolong network lifetime, instead of simply minimizing the hop count between source and destination.

In this chapter, a model for the energy consumption of wireless network interfaces is illustrated, based on energy consumption measurements. Then, new metrics for the energy evaluation of the performance of a routing protocol are presented. Finally, the state of the art of energy-aware metrics and mechanisms for ad-hoc networks is depicted.

4.1 Network Interface Energy Consumption Model

A wireless network interface can be in one of the following four states: Transmit, Receive, Idle or Sleep. Each state represents a different level of energy consumption (Figure 4.1).

- Transmit: node is transmitting a frame with transmission power P_{tx};
- Receive: node is receiving a frame with reception power P_{rx}. That energy is consumed even if the frame is discarded by the node (because it was intended for another destination, or it was not correctly decoded);

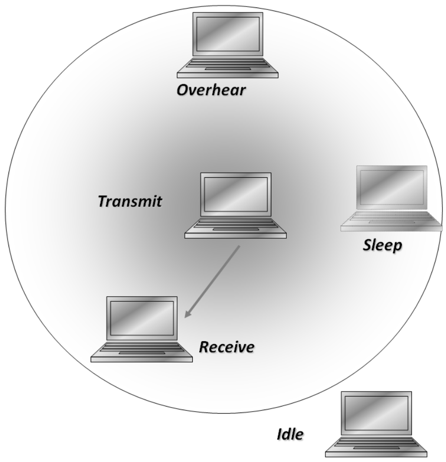

Fig. 4.1. Energy consumption in a wireless network

- Idle (listening): even when no messages are being transmitted over the medium, the nodes stay idle and keep listening the medium with P_{idle};
- Sleep: when the radio is turned off and the node is not capable of detecting signals. No communication is possible. The node uses P_{sleep} that is largely smaller than any other power.

In Table 4.1, typical values of consumption for a wireless interface (measured for a Lucent Silver Wavelan PC Card, [16]) are reported.

Table 4.1. Power value in each radio state

State	Power value
Transmit	$P_{tx} = 1.3W$
Receive	$P_{rx} = 0.9W$
Idle	$P_{idle} = 0.74W$
Sleep	$P_{sleep} = 0.047W$

The energy dissipated in transmitting (E_{tx}) or receiving (E_{rx}) one packet can be calculated as:

$$E_{tx} = P_{tx} \times Duration \tag{4.1}$$

$$E_{rx} = P_{rx} \times Duration \tag{4.2}$$

where *Duration* denotes the transmission duration of the packet.

The energy needed to transmit a packet p from node n_i can be written as $E_{tx}(p, n_i) = i \cdot v \cdot t_p$ Joules, where i is the current (in Ampere), v the voltage (in Volt), and t_p the time taken to transmit the packet p (in seconds). In our simulations, the voltage v chosen is $5V$ and we assume that the packet transmission time is calculated by $\left(\frac{p_h}{6\cdot10^6}\right)+\left(\frac{p_d}{54\cdot10^6}\right)$ seconds[1], where p_h is the packet header size in bits and p_d the payload size.

When a sender transmits a packet to the next hop, because of the shared nature of wireless medium, all its neighbors receive this packet even it is intended to only one of them. Moreover, each node situated between transmitter range and interference range receives this packet but it cannot decode it. These two problems generate loss of energy. So to compute the energy dissipated by one transmission, we must take into account these losses as follows ([17]):

$$E(p, n_a) = E_{tx}(p, n_a) + E_{rx}(p, n_b) + (n-1)\cdot E_O(p, n_i) \tag{4.3}$$

where E_{tx} , E_{rx} , and E_O denote the amount of energy spent to transmit the packet from node n_a, to receive the packet at node n_b and to overhear the packet, respectively. N represents the average number of neighbouring nodes affected by a transmission from node n_a . Equation 4.3 implies that when the network is denser, packet overhearing causes more energy consumption.

4.2 Energy performance evaluation

In order to evaluate the performance of different protocols from the energy point of view, several parameters can be calculated. Those parameters, focused on the energy behavior of the mobile networks, are listed and explained in details in the following.

- Number of alive nodes over time: this parameter evaluates the effects of the protocol on the nodes lifetime;
- Connections duration: this metric measures the lifetime of data connections between nodes;
- Nodes average energy over time: this parameter shows the behavior of energy consumption in the network;
- Nodes final energy: this metric evaluates the residual amount of energy in the network at the end of the simulation;

4.3 Related Works

4.3.1 Energy-aware metrics

The majority of energy efficient routing protocols for MANET try to reduce energy consumption by means of an energy efficient routing metric, used in

[1] In this thesis, all mobile nodes are assumed to be equipped with an IEEE 802.11g network interface card, with data rates of 54 Mbps

routing table computation instead of the minimum-hop metric. This way, a routing protocol can easily introduce energy efficiency in its packet forwarding. These protocols try either to route data through the path with maximum energy bottleneck, or to minimize the end-to-end transmission energy for packets, or a weighted combination of both.

MTPR

A first approach for energy-efficient routing is known as MTPR (Minimum Transmission Power Routing, [18]). That mechanism uses a simple energy metric, represented by the total energy consumed to forward the information along the route. This way, MTPR reduces the overall transmission power consumed per packet, but it does not directly affect the lifetime of each node (because it does not take into account the available energy of network nodes). However, minimizing transmission energy only differs from shortest-hop routing if nodes can adjust transmission power levels, so that multiple short hops are more advantageous, from an energy point of view, than a single long hop ([19]). In 802.11 we do not have access to this capability, so that, in a fixed transmission power context, this metric corresponds to a Shortest Path routing.

MBCR

Another routing metric, minimizing a function of the remaining battery power of the nodes in a path, is called MBCR (Minimum Battery Cost Routing, [18]). The proposed battery cost function is

$$f_i(t) = \frac{1}{c_i(t)} \tag{4.4}$$

where $c_i(t)$ is the battery capacity of node n_i at time t. The less capacity a node has, the more reluctant it is to forward packets.

MMBCR

If only the summation of battery costs on a route is considered, a route containing nodes with little remaining battery capacity may still be selected. MMBCR (Minimum Maximum Battery Cost Routing, [18]), defines the route cost as

$$R(r_j) = \max_{\forall n_i \in r_j} f_i(t) \tag{4.5}$$

The desired route r_O is obtained so that

$$R(r_O) = \min_{r_j \in r_*} R(r_j) \tag{4.6}$$

where r_* is the set of all possible routes.

CMMBCR

Since MMBCR considers the weakest and crucial node over the path, a route with the best condition among paths impacted by each crucial node over each path is selected. CMMBCR metric (Conditional MMBCR, [18]) attempts to perform a hybrid approach between MTPR and MMBCR, using the former as long as all nodes in a route have sufficient remaining energy (over a threshold) and the latter when all routes to the destination have at least a node with less energy than the threshold.

MDR

Power saving mechanisms based only on the remaining power cannot be used to establish the best route between source and destination nodes. If a node is willing to accept all route requests only because it currently has enough residual battery capacity, too much traffic load will be injected through that node. In this sense, the actual drain rate of power consumption of the node will tend to be high, resulting in an unfair sharp reduction of battery power. To address the above problem, the MDR (Minimum Drain Rate, [20]) mechanism can be utilized with a cost function that takes into account the drain rate index (DR) and the residual battery power (RBP) to measure the energy dissipation rate in a given node.

In the MDR mechanism, the ratio

$$f_i(t) = \frac{RBP_i(t)}{DR_i(t)} \tag{4.7}$$

at node n_i, calculated at time t, indicates when the remaining battery of node n_i will be exhausted, i.e., how long node n_i can keep up with routing operations with current traffic conditions. Therefore, the maximum lifetime of a given path r_j is determined by the minimum value of $f_i(t)$ over the path. Finally, the MDR mechanism is based on selecting the route r_O, contained in the set of all possible routes between the source and the destination r_*, having the highest maximum lifetime value.

Since the drain rate is calculated at regular time intervals, its measure is affected by isolated consumption peaks (both positive or negative). To avoid the use of incorrect values of drain rate during these peaks, an α parameter can be introduced. This parameter makes the drain rate value between adjacent intervals smoother, acting in the following manner: after calculating the drain rate sample at interval i, $DR_{sample}(i)$, MDR uses a value of drain rate of

$$DR(i) = (1 - \alpha) \cdot DR_{sample}(i) + \alpha \cdot DR(i-1) \tag{4.8}$$

MDR suffers from the same problem as MMBCR, ignoring the total transmission power consumed by a single path: this way, it could even lead to a higher overall energy consumption in the network. To prevent this issue, MDR can be introduced in a hybrid way, as a CMDR (Conditional MDR) metric: as far as all nodes in a route have sufficient remaining lifetime (over a threshold), a simple MTPR approach is used.

Other works (like [21]) use a larger number of variables in the cost function of the algorithms, for example by taking into account not only the residual energy and the transmission power, but also the energy cost of possible packet retransmissions. Similarly to the MDR metric, an important aspect for the design of energy aware routing protocols is highlighted: the estimation of future energy consumption. The energy that is expected to be used in order to successfully send a packet across a given link is estimated by a cost function that comprises both a node-specific parameter (battery power $B(i)$ of node i) and a link-specific parameter (packet transmission energy $E(i,j)$). The cost of the reliable communication across the link (between nodes i and j) is defined as

$$C(i,j) = B(i) \cdot E(i,j) \tag{4.9}$$

The expected transmission energy is defined by the power needed to transmit a packet over the link between nodes i and j ($T(i,j)$) and the link's packet error probability ($p(i,j)$):

$$E(i,j) = T(i,j) \cdot (1 - p(i,j)) \cdot L \tag{4.10}$$

The main reason for adopting the above is that link characteristics can significantly affect energy consumption and can lead to excessive retransmissions of packets. The cost of choosing a particular link is defined as the maximum number of packets that can be transmitted by the transmitting node over that specific link. It is also assumed that there is complete absence of any other cross traffic at that node. The maximum lifetime of a given path is determined by the weakest intermediate node.

Another approach ([22]) make use of the available battery capacity by means of battery-sensitive routing. That approach studies the lifetime of the battery and proposes an algorithm based on two processes, namely, recovery (reimbursement) and discharging loss (over-consumed power). These processes are experienced when either no traffic or new traffic is transmitted. This study led to the design of a cost function that penalizes the discharging loss event and prioritizes routes with “well recovered” nodes. Thus, battery recovery can take place and a node's maximum battery capacity can be attained. The selection function is a minimum function over the cost functions of all routes.

4.3.2 Energy saving techniques at routing layer

The problem of energy efficiency in MANETs can be addressed at different layers. In recent years, many researchers have focused on the optimization of energy consumption of mobile nodes, from different points of view. Some of the proposed solutions try to adjust the transmission power of wireless nodes ([23], [24]). Other proposals tend to efficiently manage a sleep state for the nodes: these solutions range from pure MAC-layer solutions (as the power management of 802.11) to solutions combining MAC and routing functionality ([25]). Finally, there are many proposals which try to define an energy efficient routing protocol, capable of routing data over the network and of saving the battery power of mobile nodes ([26], [27], [20], [28], [29], [18], [30]). Such proposals are often completely new, while others aim to add energy-aware functionalities to existing protocols, like AODV ([31], [32]), DSR ([33], [34]) and OLSR ([36], [37], [38]).

The aim of energy-aware routing protocols is to reduce energy consumption in transmission of packets between a source and a destination, to avoid routing of packets through nodes with low residual energy, to optimize flooding of routing information over the network and to avoid interference and medium collisions.

Some routing protocols organize wireless nodes into clusters, such as Leach ([39]). In ([40]) the conditions under which such protocols are energy efficient are established and the optimal radius of a cluster is determined.

Existing energy efficient routing protocols can be first distinguished by the number of paths maintained to a destination: a single path or multiple paths.

Multipath routing protocols ([41], [42]) have the advantage of sharing load of any flow on several paths, leading to a lower consumption on the nodes of the selected paths. It has been shown in ([43]) that two paths with different links are generally sufficient.

We can distinguish three families of energy efficient routing protocols:

- the protocols selecting the path consuming the minimum energy. The advantage is that each transmission of a packet from its source to its destination minimizes the energy consumed. We can cite for example ([32]) and a more sophisticated protocol ([35]) where the selected path minimizes the additional energy dissipated by the routing of the new flow, taking into account the SINR and the energy lost in interferences. However, such protocols use always the same nodes (those minimizing the energy consumed) without any consideration on their residual energy. Consequently, these nodes will exhaust their battery more quickly than the others and the network lifetime is not maximized.
- the protocols selecting the path visiting the nodes with the highest residual energy, such as ([44]). Each flow is ensured to have enough energy on the selected path: depleted nodes are avoided. However, the path selected does not minimize the energy needed to transmit a flow packet from its source to its destination. Hence, the network lifetime may not be maximized.
- the hybrid protocols selecting the path with the minimum cost, where the cost takes into account the residual energy of each visited node (and possibly its neighbors) and the energy consumption of a packet on this path. These protocols avoid the problems encountered by the protocols

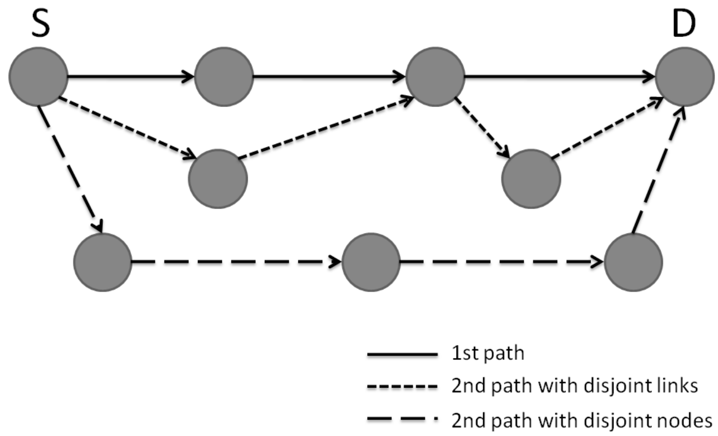

Fig. 4.2. Multipath Routing

of the two previous categories by weighing the factors used in the cost computation. We can cite for instance ([45]).

Proactive energy-aware routing

The energy optimization of a proactive routing protocol can exploit various network layer mechanisms, like control information forwarding. In OLSR, for example, the MPR selection mechanism can be varied in an energy-aware way. As suggested in RFC 3626, MPRs can be selected by their residual energy, rather than by their 2-hop neighborhood coverage ([37]). Some works applied both techniques (MPR selection criteria modification and path determination algorithm modification) to increase the energy efficiency of OLSR protocol ([36], [46], [19]).

Another mechanism that allows energy saving in OLSR protocol (without changing its behavior) is the Overhearing Exclusion ([46]). Turning off the device when a unicast message exchange happens in the node's neighborhood, can save a large amount of energy. This can be achieved using the signaling mechanisms of the lower layers (i.e. the RTS/CTS exchange performed by IEEE 802.11 to avoid collisions), and does not affect protocol performance. In fact, OLSR does not take any advantage from unicast network information directed to other nodes (while other protocols, such as DSR, have mechanisms to do so).

Reactive energy-aware routing

The Local Energy-Aware Routing (LEAR, [47]) algorithm grants each node in the network permission to decide whether to participate in route searching: this way, the decision process is spread among all nodes in the network. That algorithm uses the energy profile of the nodes as a main criterion for the routing decision. The residual energy of each node defines the reluctance or willingness of that node to reply to route requests and forward data traffic. When energy E_i in a node i is lower than a given threshold Th

$$E_i < Th \tag{4.11}$$

the node does not forward the route request control message, but simply drops it. Thus, it will not participate in the selection and forwarding phase.

The technique of spreading the responsibility from the source/destination nodes to the intermediate nodes avoids the needing for a periodic exchange of control information, thus leading to reduced bandwidth and energy consumption. This technique has been commonly used to improve the performance of the routing protocols in many recent approaches.

Hybrid energy-aware routing

The work in [25] introduces a new way of optimizing the energy consumption in a wireless network, independently from the routing protocol adopted by the nodes. Assuming that all the devices in the network are equipped with a GPS (Global Positioning System) receiver, that work introduces the Geographical Adaptive Fidelity (GAF) for ad-hoc wireless networks. GAF conserves energy by identifying nodes that are equivalent from a routing perspective and then turning off unnecessary nodes, keeping a constant level of routing fidelity. GAF moderates this policy using application- and system-level information; nodes that source or sink data remain on and intermediate nodes monitor and balance energy use. Simulations of GAF suggest that network lifetime increases proportionally to node density. Power consumption in current wireless networks is idle-time dominated, so GAF focus on turning the radio off as much as possible.

4.3.3 Comparative Performance Evaluation from an Energetic Point of View

Many energy efficient routing protocol proposals were originally studied for sensor networks, where the limited energy of nodes is a strong constraint; in MANET, however, the requirements are different: a node has generally more hardware resources (capable of better performance, but consuming more energy) and the protocol must preserve the resources of every node in the network (not only a subset of them, because each node can be, at any time, source or destination of data). A single node failure in sensor networks is usually unimportant if it does not lead to a loss of sensing and communication coverage; ad-hoc networks, instead, are oriented towards personal communication and the loss of connectivity to any node is significant.

The lifetime of a network is usually defined according to the following criteria ([48]):

- the time until the first node burns out its entire battery budget;
- the time until a certain proportion of the nodes fails;
- the time until network partitioning occurs

A single node failure represents a serious problem in ad-hoc networks, because its occurrence can lead to the network partitioning. In contrast, a single node failure in sensor networks is usually unimportant if it does not lead to a loss of sensing and communication coverage. Ad hoc networks are oriented towards personal communications and the loss of connectivity to any node is significant. Consider, for example, a disaster recovery scenario. In such case, it is important that the rescuers do not lose connectivity with any other member of their team, and the connectivity among rescuers should be maintained as long as possible, or at least the duration of the rescue operation. Network partitioning interrupts communication sessions and can be caused by node movement or by node failure due to energy depletion. While the former cannot be controlled by the routing protocol, the latter can be avoided through appropriate routing decisions.

Operational lifetime can be defined as the time until network partitioning occurs due to battery outage. In order to achieve the objective of maintaining connectivity as long as possible, the distribution of network tasks among its nodes should be equal so that they all decrease power at the same rate and eventually run out of energy at approximately the same time. The network must be designed to achieve the simultaneous failure of the nodes (due to a lack of energy), so that communication requirements are met. This leads to consider as the operational lifetime of such networks their relative lifetime, rather than the absolute lifetime of their devices. The useful lifetime of ad-hoc networks can be significantly lower than the network's devices lifetime, but from an engineering and application perspective the former time span is much more interesting and meaningful. For instance, a case could be envisaged in which some nodes have fully charged batteries but are unable to establish successful communications because they belong to disconnected parts of the network or must communicate with nodes that are turned off due to a lack of energy. In such a scenario, the absolute lifetime of a network will be longer compared to the useful life span, but this is not of practical interest.

Many works have been presented in literature to give a measure of the energy consumption of various routing solutions in a wide range of scenarios, exploring the behavior of different protocols (especially OLSR and DSR) and trying to highlight the strength and weakness points of each of them ([2], [49],

[50]). These researches are a good starting point for every energy-aware routing proposal for MANETs. To achieve the desired behavior, some proposals make use of clustering ([39]) or maintain multiple paths to destinations (in order to share the routing load among different nodes, [43]).

4.3.4 Scalable energy-aware routing

In a hierarchical network, the network elements are partitioned into several groups, called clusters. In each cluster, there is a master node that manages all the other nodes (slave nodes) within the cluster. The depth of the network can vary from a single tier to multiple tiers. However, most hierarchical networks are two-tier networks.

Two-tier mobile ad hoc networks require sophisticated algorithms to perform clustering based on limited resources, such as the energy of each node, to communicate with each other. The cluster area of a node is related to the transmission power. Therefore, a larger cluster area requires more energy. The energy required by a two-tier mobile ad hoc network varies with the clustering configuration (the master node selection of slave nodes) because the transmission power of each node must be set to satisfy the minimum power level at the receiving node.

Therefore, there exists an optimum clustering configuration that minimizes the call drop rate and the energy required for the still snapshot of the network. However, the optimum clustering configuration cannot be calculated quickly. A heuristic clustering scheme resulting in energy conservation for the network that can be implemented and executed in a limited time is needed for real-time clustering.

In [51] the authors propose two distributed heuristic clustering schemes for energy conservation in two-tiered mobile ad hoc networks. The proposed schemes can be implemented and executed in real time. The mean transmission power and the call drop rate for the proposed schemes approximate optimum results. Hence, the proposed schemes are suitable for periodic or event-driven cluster reconfiguration. The proposed double-phase scheme is useful when energy conservation and call completion are more important than computing power and the speed of the scheme. In the opposite case, the proposed single-phase scheme can be adopted.

4.3.5 Implementation issues in energy management functionalities

Aiming to extend the time until network partitioning, routing protocol designers often try to optimize the use of battery power, in order to maximize the life of a node. However, extending nodes' lifetime could not be the better way to increase the connectivity between all of the nodes in the network.

The min-max algorithms are implemented to overcome the problem that arises when the total energy cost of routes is used as an argument for the selection of a route, that is, when nodes with low residual energy are excluded.

However, if these protocols are analyzed in terms of a network's operational lifetime, the problem of extending the network's lifespan for as long as possible persists. Simulation results (like in [52]) show that protocols that implement min-max algorithms or the energy drain rate have lower values for the standard deviation of the remaining energy in comparison with algorithms that use transmission power as a metric. Furthermore, the distribution of the energy of the nodes along the path is not even in any of the protocols. If in the cost function it is taken into account only the specific energy state of the nodes without considering the overall distribution of the energy along the routes, optimal results will not be obtained when the operational lifetime of a network is being examined.

The energy-aware protocols usually implement only energy-wise metrics. An improvement on this general approach is the inclusion of the speed with which the battery is burned. The energy drain rate is helpful in stopping a node from powering down. It does so by deviating traffic when a certain threshold is reached. The load at each node and in its neighbors is an indicator of the energy to be consumed for transmitting packets by a particular node. Moreover, it accounts for the shared nature of the radio as a medium. The network tasks in which each node is involved are a main item in the battery budget. When this item is considered along with the current energy state of a node, it can regulate the speed of energy consumption.

Additional metrics should be considered, such as the fact that when neighboring nodes are engaged in transmitting packets, they are competing for the wireless medium. Retransmissions that may possibly take place ([21]) should also be taken into consideration. The resulting collisions and retransmissions are energy-consuming and cannot simply be represented by the residual energy metric.

Part II

Personal Contribution

5

Research Methodology

This chapter outlines the research methodology used in this research work to study the performance of routing protocols on MANETs. The chapter is divided into three sections. Section 5.1 presents the research paradigms and methodology used. Section 5.2 outlines the tools used in carrying out the experimental research. Finally, Section 5.3 illustrates the methodology adopted for the analysis of the results obtained in simulations.

5.1 Research Paradigm

Based on networking and its related areas, methodologies used for modelling and performance evaluation of routing protocols are analytical modelling, direct or real experiment and computer simulation.

Analytical modelling is based on mathematical computation and analysis. Although analytical analysis may found to be a good ground for formulating new routing protocols, its weakness lies in operating and controlling protocols. Furthermore analytical modelling cannot represent the dynamic nature of MANETs, its configuration and reconfiguration for large networks would be too troublesome.

Real experiment, however, has an advantage in obtaining fairly accurate results as the research is carried out in reality, where the influences and routing behaviours can be observed depending on the surrounding environment. Simple levels of abstraction in simulation cannot provide a solid base validation of routing protocol behaviour as compared to real experiment. The complexity of MANETs, however, could not be simply tested in real experiments due to the high cost and the complex nature of mobile ad hoc networks, taking numerous efforts and resources to carry out the experiments and performance evaluations.

A simulation experiment is defined as a process of constructing models of a system consisting tho object of the study and then conducting simulation experiments with the model using a computer program (also referred to as simulation tool or simulator) to help solve or find a solution to the problem. The aim, when using any simulator, is to accurately model and predict the behaviour of a real world problem in a system.

Computer simulation is used to generalise measurement results, verify analytical models, compare the performance of existing protocols and also evaluate the performance of new protocols. However a potential problem exists in simulations where simulators are not generating accurate or representative results. An extensive knowledge about a good simulator is required to overcome this problem. A good simulator should be easy to use, flexible in model development and in its modification and validation and should incorporate appropriate tools for the analysis of results.

Computer simulation methodology is used to carry out the proposed research. Usually, the performance of routing protocols is evaluated using computer simulation techniques which, unlike analytical methods, use fewer assumptions and behave more like real systems. The complexity of routing algorithms is another strenght point of simulation methodology. In addition, simulation offers more flexibility in model development, validation, and performance evaluation.

5.2 The ns-2 Simulator

In this research work, the simulation software adopted for the evaluation of existing and proposed routing solutions is the ns-2 (Network Simulator 2, [53]), developed and maintained by the VINT Project (Virtual InterNetwork Testbed, [54]).

5.2.1 Features

ns-2 is a discrete event, object oriented simulator, written in C++; it uses OTcl (Object Tool Command Language) as a command and conguration interface. The simulator supports a class hierarchy in C++ (also called the compiled hierarchy), and a similar class hierarchy within the OTcl interpreter (also called the interpreted hierarchy). The two hierarchies are closely related to each other; from the users perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy.

5.2.2 Mobile node organization

A mobile node (Figure 5.1) is composed by a number of objects that realize, in a semplified way, the typical ISO-OSI structure of a real network node. The connection between nodes, in the software, is performed by a set of pointers (“targets”) that provide the references used to communicate with a module or to send packets to a neighbor node. In the following the modules of a mobile node are described.

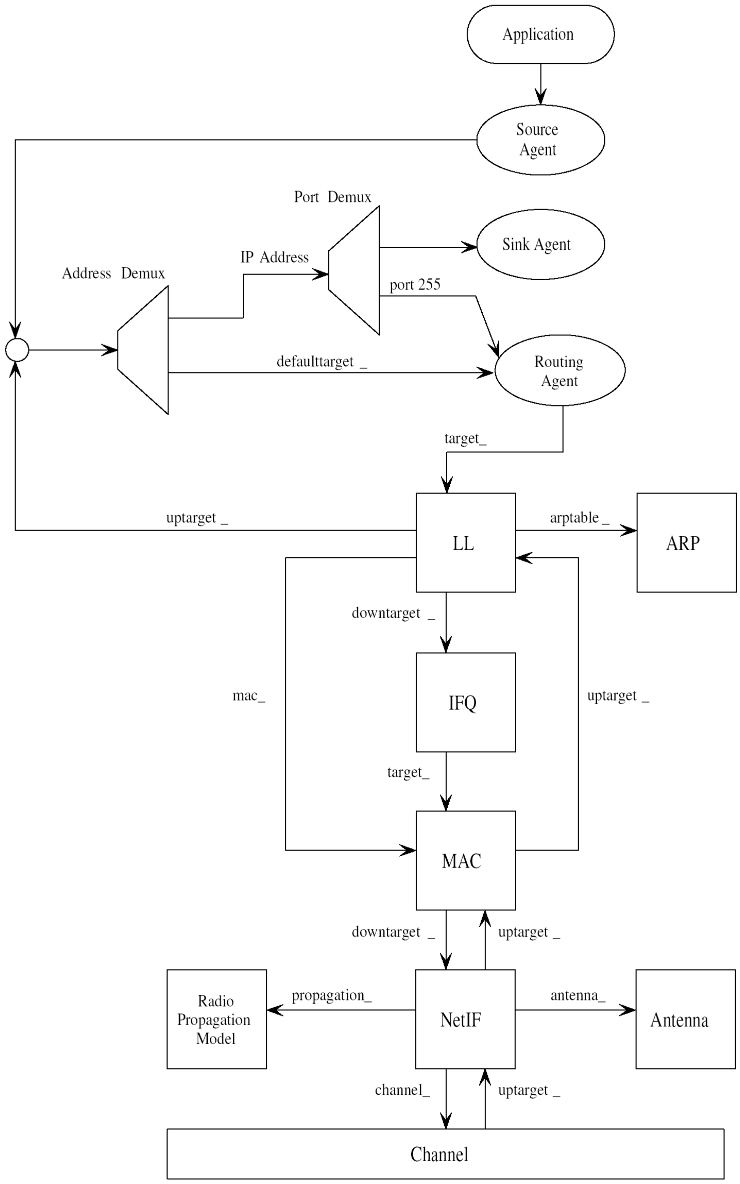

Fig. 5.1. Mobile node structure in ns-2 simulator

- Propagation model - calculate the power of signal detection, applying a propagation model (Free Space, Two Ray Ground);
- Network InterFace (NetIF) - implements the interface between the node and the communication channel and defines the reception thresholds;
- Antenna - defines a simple, omni-directional antenna;
- InterFace Queue - manages the packet queues;
- Mac - implements the Medium Access Control protocol (for example, IEEE 802.11);
- LL and ARP - implement Address Resolution Protocol mechanisms;
- Routing Protocol - manages the data packet routing procedures;
- Agent - implements the transport protocol (like TCP or UDP);
- Application - controls the generation of packets (dimension, frequency, distribution).

5.2.3 Events, scenarios and connections management

ns-2 is an event driven simulator: each module, in order to manage particular situations (like packet receptions or timeouts), builds an "Event" object. For each generated event, its execution time is calculated and a pointer to an handler is setted to it. The handler contains the instructions for the management of the event. The simulator Scheduler inserts and removes the events in a dynamic data structure. The Scheduler, furthermore, updates the important "clock" variable. It represents the virtual clock of the simulation, which the simulator object can access to.

A simulation scenario consists of a sequence of instructions, specifying the position of each node in the space and the connections beetween nodes. The simulator provides some tools for the simulation scenario generation. The Setdest program, for example, creates a configuration file containing the nodes cohordinates and the instructions defining nodes movements (speed, direction, pause time).

For the connections generation, the Cbrgen program builds automatically a random list of source-destination pairs, defining the start time for their communications. The generated traffic is of CBR (Constant Bit Rate) type: that means that the source node, after setting the connection, continuously sends packets to the destination, with the defined send rate.

5.2.4 Energy model

The Energy Model module is an optional attribute of the ns-2 mobile node. It traces the residual energy of the node and decrements it after specific simulation events (transmission, reception, idle time). ns-2 provides a simple energy model, that saves only the information about the initial energy of the node and about its current energy.

For this research work, a more complex energy model has been developed and added to ns-2. That energy model maintains the information about the energy consumption of the node divided by type (transmission, reception, overhearing and idle). That way, more accurate statistics can be extracted from a simulation, regarding the energy consumption of the node.

An example of Tcl script for the simulation of our proposal is shown in Appendix A. The Appendix B, instead, contains an example of the Perl scripts used for the extraction of the evaluation statistics from the tracefiles generated by the simulator.

5.3 Analysis of the Results

As seen before, a simulation represents the evolution of a complex model, influenced by a number of random conditions. The main source of randomness in a simulation is given by the random movements of the nodes in the simulation scenario. For that reason, is preferable to repeat a simulation for a certain number of times, to have more reliable statistics about the simulation results.

To estimate the precision and the reliability of the simulation results, we can use stochastic methods. For example, if we want a reliable value of a random variable X (a performance index), we can make n repetitions of the simulation experiment, obtaining n independent observations of the random variable $(X_1, X_2, \ldots, X_n)$.

If we suppose that X has an average value μ and a variance σ^2, an estimation of μ is given by:

$$\bar{X}(n) = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{5.1}$$

In general, however, $\bar{X}(n) \neq \mu$.

The confidence interval method consists in the determination of an interval around the value of $\bar{X}(n)$, predicting this way the probability (confidence) that μ is contained in that interval:

$$Pr\{|\bar{X}(n) - \mu| < \delta\} = 1 - \alpha \tag{5.2}$$

where δ is the half of the amplitude of the confidence interval. Usually, $1 - \alpha$ takes the values 0.9, 0.95 or 0.99 (90, 95 or 99% of reliability).

The variance of random variable X, instead, can be calculated from:

$$S^2(n) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X}(n))^2 \tag{5.3}$$

The variance of $\bar{X}(n)$ is:

$$Var\left[\bar{X}(n)\right] = \frac{\sigma^2}{n} = \frac{S^2(n)}{n} = \frac{1}{n(n-1)}\sum_{i=1}^{n}(X_i - \bar{X}(n))^2 \tag{5.4}$$

If the number of independent experiments is high ($n > 30$), we can assume that $\bar{X}(n)$ has a Gaussian distribution (theorem of the central limit). But, if the measurements X_i have a normal distribution, the variable

$$t_n = \frac{\bar{X}(n) - \mu}{\sqrt{S^2(n)/n}} \tag{5.5}$$

has a distribution called *t-Student*, with $n - 1$ freedom degrees, and in this case the confidence interval can be exactly expressed as:

$$\delta = t_{n-1,1-\alpha/2}\sqrt{S^2(n)/n} \tag{5.6}$$

Some values of the *t-Student* distribution are listed in Table 5.1.

Table 5.1. t-Student distribution values table

	$1 - \alpha/2$		
n-1	**0.9**	**0.95**	**0.975**
1	3.078	6.314	12.706
2	1.886	2.920	4.303
3	1.638	2.353	3.182
4	1.533	2.132	2.776
5	1.476	2.015	2.571
6	1.440	1.943	2.447
7	1.415	1.895	2.365
8	1.397	1.860	2.306
9	1.383	1.833	2.262
10	1.372	1.812	2.228

6

Energy Optimization Proposal

The proposal of this work consists in some different energy-aware improvements for classical OLSR protocol, as proposed in RFC 3626 ([12]), in order to increase energy robustness without loss of performance. The battery status of the nodes in the network has been taken into account, aiming to prolong the lifetime of the nodes and to maximize the overall throughput of the energy-constrained mobile network. These effects are obtained changing the routing metric from the minimum-hop count of classical OLSR to one of the most promising energy-aware routing metrics proposed in literature, the Minimum Drain Rate (MDR) metric ([20]):

6.1 Energy-aware Metric for OLSR

Compared with other energy-aware routing metrics known in literature (based on remaining battery capacity of the mobile nodes, [18]), MDR gives a more accurate measure of energy consumption (predicting the remaining lifetime of each node by the knowledge of its energy consumption over time. To adopt this metric within the OLSR protocol, the local Drain Rate calculation was implemented in the OLSR agent, the `TC` packet format was changed (in order to propagate the energy information over the network) and the routing table calculation was reformulated (to take account of the energy information obtained from the protocol).

Moreover, to avoid some problems due to the use of an energy-aware metric (like the rise of overall energy consumption in the network, caused by non-optimal paths selected to preserve the most stressed nodes) a conditional approach was introduced, called CMDR (Conditional MDR). In CMDR, the energy-aware metric is used for path selection only if at least one node, in every possible path to the destination, goes under a particular value of predicted lifetime (namely, the γ parameter).

In Figure 6.1 the behavior of CMDR routing metric is explained by means of an example. The values in the nodes represent the predicted lifetimes, and the γ threshold is set to 5: when at least one node in a path predicts a lifetime under the threshold, the node of the path with the minimum value of predicted lifetime is highlighted (it is the critical node of the path); otherwise, only the hop count of that path matters for routing calculation. In the first case (A), the central and lower paths will be preferred to the higher one (as they do not present any node under the threshold), and the central path will be selected by CMDR because it has less hops between the source and destination. In the second case (B), every path to the destination has at least one node with a lifetime shorter than the γ parameter: this time, the lower path will be selected by CMDR metric (even if it results longer than the central one in terms of hop count), because it presents a higher value for the predicted lifetime of its critical node.

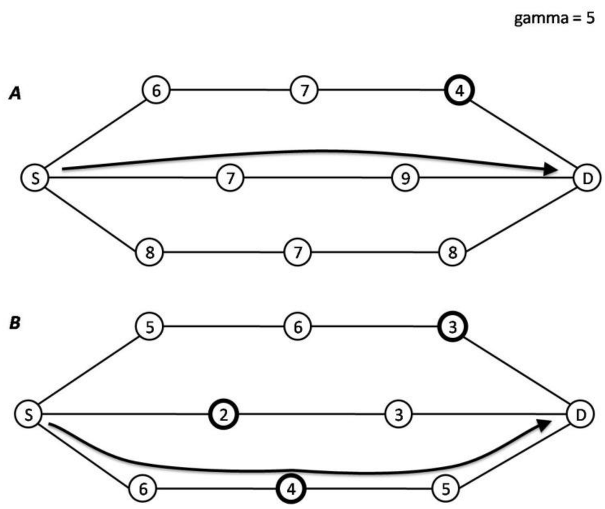

Fig. 6.1. Example of CMDR routing metric behavior

In a similar case, a minimum hop count routing metric would exploit the shortest path to deliver data between the source and the destination, until a node on this path ceases to function (because it exhausted all its battery charge). As soon as it happens, the protocol has to compute a new path (possibly causing data loss, until the old path breaking is detected and the new path is available).

The previous example clearly illustrates the advantages of using a routing aware metric in an energy constrained mobile network, especially in the bigger, denser and more loaded ones (where the availability of multiple paths can be exploited to avoid using only a small subset of nodes in the network for traffic delivery).

In terms of pseudo-code, the CMDR metric for routing table computation has been implemented as follows:

Routing table computation with Conditional-MDR metric

```
r_tab.clear();
default_metric = MAX_LT;
for(n in neighbor_set)
    r_tab.add(n, n, default_metric);
do{
    for(t in topology_set){
        dest=r_tab.lookup(t.dest);
        last=r_tab.lookup(t.last);
        if(last != null){
            new_metric = min(last.metric, t.metric);
            if(new_metric > gamma)
                new_metric = MAX_LT;
            if(dest == null OR new_metric > dest.metric){
                r_tab.add(dest, last.next, new_metric);
                added = true;
            }
        }
    }
} while(added)
```

The heuristic used to calculate the routing table is the same proposed in RFC 3626: when a change in network topology is detected, all the existing entries are deleted, and a brand new routing table is calculated by the node, based on the information stored locally in the topology control repository. Firstly, all the neighbors in the neighbor set are added as routing entries. Then, the topology set is evaluated, and a new entry is added every time a new path (or a path with a lower cost than the existing one) is found from a topology tuple. When no new paths are added by the analysis of the topology set, the heuristic stops and a new routing table is ready for the node.

The energy metric used to evaluate the paths is the minimum predicted lifetime in the path, as stated below. This value is propagated over the network in the Topology Control messages, defined in the OLSR protocol. In order to spread such information without charging the protocol with more control information (since OLSR is a proactive routing protocol, and it causes a significant overhead in the network), the reserved field of RFC 3626 specification for OLSR was used. Our implementation, in fact, filled the reserved field of `TC` message (2 bytes) with the predicted lifetime of the originator node, expressed in remaining seconds. Using this field, the maximum value of lifetime that a node can claim (MAX_LT) is 65535 seconds (corresponding to about 18 hours): the default value for the energy metric in routing calculation was set to that value.

The value of the predicted remaining lifetime (PLT) is regularly calculated by each node (namely, at every `HELLO_INTERVAL`), from the current value of energy drain rate at the node:

$$PLT = \min\left(\frac{current\_energy}{DR}, MAX\_LT\right)[sec] \tag{6.1}$$

The value of the energy drain rate (DR) is calculated according to the MDR metric definition ([18]):

$$DR = \alpha \cdot dr_{old} + (1-\alpha) \cdot dr_{sample}\,[J/s] \tag{6.2}$$

where dr_{sample} represents the actual drain rate, calculated in the last interval, dr_{old} is the previous value of drain rate for the node and α is a value between 0 and 1.

6.2 Energy-aware Mechanisms for OLSR

Another energy-aware improvement to the OLSR protocol consisted in the Energy-Aware Willingness setting: a modification to the algorithm of willingness calculation of OLSR nodes, aiming to introduce energy parameters in the MPR selection process. Formally, no changes to the OLSR MPR selection heuristic (as proposed in RFC 3626) were introduced. Instead, the existing concept of the nodes willingness was exploited. Since the willingness of a node (namely, its availability to transport data on behalf of other nodes) is used in OLSR to select which neighbors add to the MPR set, employing an energy-aware metric in willingness setting permits to automatically inject energy-awareness in MPR selection mechanism.

This work adopts both predicted lifetime and remaining battery power to set the willingness of a node. The heuristic consists in calculating the actual level of both values for the node ("short", "medium" or "long" for the former, "low" or "high" for the latter), and mapping each of their six possible combinations to a willingness value. OLSR defines 8 values of willingness that a node can claim, from `WILL_NEVER` ("this node should never be selected as an MPR") to `WILL_ALWAYS` ("this node must be an MPR for every neighbor"). In this work, only two values of willingness are used beyond `WILL_DEFAULT` (the only one used in RFC 3626): `WILL_LOW` (for nodes in bad energy status) and `WILL_HIGH` (for nodes in good energy status).

The heuristic adopted by this work is graphically illustrated in Figure 6.2.

The six lines depicted here represent the six possible battery consumption behaviors of a node, depending on its remaining battery and predicted lifetime. The energy status can vary from high battery level and slow drain rate to low battery power rapidly exhausting. The figure illustrates that a node is considered in a bad energy status if its predicted lifetime is lower than a threshold ($SHORT\_LT$), or if it has a lifetime under another threshold

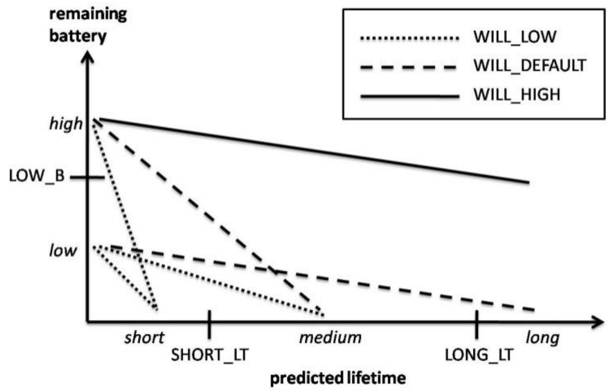

Fig. 6.2. Energy-aware willingness setting heuristic

(*LONG_LT*) and a low remaining battery charge (under a third threshold, namely *LOW_B*). Instead, if the node has both a high battery charge and a long predicted lifetime, it can claim a good energy status. In all the other cases, the node propagates the default value for willingness.

Table 6.1 illustrates the mapping between the (remaining_battery, predicted_lifetime) pairs and the willingness value adopted by the nodes, as have been set in this work.

Table 6.1. Willingness values adopted by Energy Aware OLSR nodes, depending on energy status

	Predicted lifetime		
Remaining battery	**short**	**medium**	**long**
low	WILL_LOW	WILL_LOW	WILL_DEFAULT
high	WILL_LOW	WILL_DEFAULT	WILL_HIGH

The following pseudo-code explains the heuristic used to calculate the willingness value for each node, in detail:

Energy Aware-Willingness heuristic

```
battery = current_energy/init_energy;
lifetime = MAX_LT;
if(drain_rate() != 0.0)
      lifetime = current_energy/drain_rate();
willingness = WILL_DEFAULT;
if(lifetime < SHORT_LT)
      willingness = WILL_LOW;
else{
```

```
        if(battery < LOW_B AND lifetime < LONG_LT)
              willingness = WILL_LOW;
        else if(battery > LOW_B AND lifetime > LONG_LT)
              willingness = WILL_HIGH;
}
```

The battery level value is calculated by the ratio between the current energy level and the initial energy of the node, while the predicted lifetime value is calculated as seen before for MDR metric. The willingness value is calculated by each node in the network at every `HELLO_INTERVAL`, and is sent to neighbors in the `HELLO` message regularly originated by the node itself. Using the willingness values received by its neighbors, every node can apply energy-awareness in its MPR selection.

6.3 MDR Tuning

The α parameter used by MDR mechanism ($0 \leq \alpha \leq 1$) influences the actual use of nodes in the network. Nodes that declare high values of drain rate will be seen as overloaded, and the routing protocol will try to avoid them. Setting $\alpha = 0$, every node will declare its real energy consumption during the last evaluation interval, and the nodes that have experimented short peaks of energy consumption in the interval will be avoided during the next interval. On the other hand, setting high values of α will lead to a slow convergence to real consumption value.

Generally, the drain rate value declared by a node at the n-th interval can be written as (according to Equation 4.8):

$$DR(n) = (1 - \alpha) \cdot \sum_{i=1}^{n} (\alpha^n - i \cdot DR_{sample}(i)) \tag{6.3}$$

If the energy consumption of the node is constant (i.e., $DR_{sample}(i) = K$), $DR(n)$ will tend to the real value of consumption, K, and the value of α will state how fast it will reach this value.

Another particular case is when the energy consumption is of "on/off" type: the node spends a constant amount of energy for a period of time (T_{on}), and does not consume energy for another period of time (T_{off}). In this case, the consumption is still constant (while oscillating), but the drain rate measure at the node depends also on the ratio between the calculation interval (of width W) and the on/off period (defined as $T = T_{on} + T_{off}$). In general, we can write:

$$\overline{DR}_{sample}(n) = \lfloor W/T \rfloor \cdot \beta \cdot T + R_1 + R_2 \tag{6.4}$$

Where

$$R_1 = \max\left(0, \beta T - \left((n-1)\,W - T\lfloor\frac{(n-1)W}{T}\rfloor\right)\right) \tag{6.5}$$

$$R_2 = \min\left(\beta T, nW - T\lfloor\frac{nW}{T}\rfloor\right) \tag{6.6}$$

$$\beta = \frac{T_{on}}{T} \tag{6.7}$$

The value $\overline{DR}_{sample}(n)$ represents a normalization of the actual drain rate value at interval n. The real value of drain rate can be obtained easily, multiplying it by the peak consumption value (in Watts), and dividing it by the window size (in seconds).

If W is a multiple of T, the residuals R_1 and R_2 go to 0, and Equation 6.4 reduces to a constant value, βW. This case is similar to the previous. Otherwise, the measured value of drain rate will be different at every interval, depending on the variable residuals, and it will oscillate around the mean value βW.

It can be immediately noticed how a constant value of α parameter can not satisfy both cases. In facts, in the former (DR_{sample} constant) a high value would be better (thus reducing convergence time), while in the latter (DR_{sample} oscillating around a fixed value) one would prefer a low value (thus reducing oscillations).

Since no fixed value of the α parameter in MDR mechanism can satisfy a wide range of energy consumption cases, we studied an adaptive model, to give a more accurate estimation of the Drain Rate. In our model, for the n-th estimation window, a normalized drain rate variation index is calculated, defined as follows:

$$\delta(n) = \min\left(\frac{|DR_{sample} - DR(n-1)|}{DR_{sample}}, 1\right) \tag{6.8}$$

This permits to consider the variation of the DR index during two consecutive interval time. Thus, if some change in the traffic occurs, the drain rate can be estimated again giving more or less weight to the α parameter in a dynamic way.

Using this index, α parameter can be dynamically set as:

$$\alpha(n) = 1 - \delta(n) \tag{6.9}$$

and the drain rate is calculated by the formula:

$$DR(n) = (1 - \alpha(n)) \cdot DR_{sample}(n) + \alpha(n) \cdot DR(n-1) \tag{6.10}$$

6.4 Geographical Optimization of OLSR

The approach presented in this section joins the advantages of OLSR protocol (that provides an accurate packet delivery, at the cost of a high control overhead and of a waste of bandwidth used to maintain routing tables) to those of geographical routing (a more lightweight approach, but less effective).

The idea is to use the proactive routing for local forwarding (taking advantage by the full knowledge of k-hop neighborhood given by OLSR mechanisms), and the geographical routing to forward packets outside the k-hop cluster (drastically reducing the overhead and improving the scalability of the network).

The geographical protocol used to apply this approach is GPSR (see Section 2.3.4).

6.4.1 Network density and cluster size

The behavior of a greedy geographical routing protocol is influenced by the density of nodes in the network. As shown in Figure 6.3, in a high density network that algorithm is more likely to minimize the path between source and destination than in a low density condition.

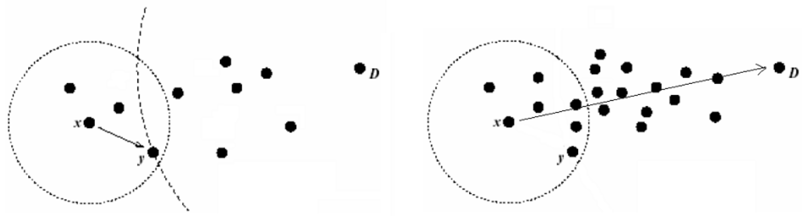

Fig. 6.3. Greedy algorithm behavior with different densities

In the proposed hybrid protocol, the local routing radius is adaptive: the cluster dimension is adapted to the source node neighbors' density. A node with a little number of neighbors will save more nodes in its routing table, inspecting the control packet coming from fartest nodes. The control information is exchanged by means of OLSR packets. In particular, a node D is said to be a local destination for a node S if the distance between the two nodes is lesser or equal to k hops. The value of the routing radius, k, is dimanicly adapted on the basis of the local node density.

For example, if the neighborood of a node has a low density, it will select an higher value of k, an it will forward the packet in a proactive way for each destination inside the zone delimited by the k-hops radius. On the other hand, a node with a denser neighborhood will forward its packets in a greedy geographical way for destinations nearer than in the previuos case.

Figure 6.4 shows the cluster dimension selection mechanism: the radius is larger when the neighbor density is smaller (node X), and it will be smaller for the nodes with an high density neighborhood (node A).

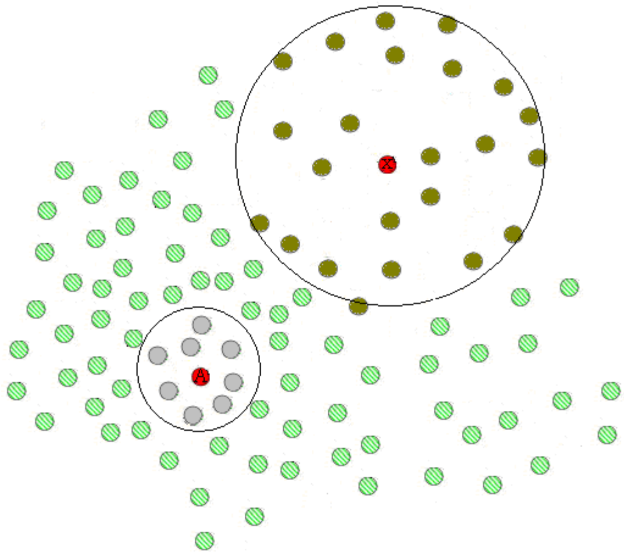

Fig. 6.4. Variable local routing radius, based on density

In a low density scenario, the proactive routing protocol is more efficient than the geographical one, while at higher nodes density GPSR performs better than OLSR.

6.4.2 Energy issues

The basic greedy parameters does not apply energy aware logic to its algorithm. It always uses the nearest node in the path to the destination to forward the packets. In a limited energy context like a MANET, it could lead to a rapid battery consumption for those nodes, until it ceases to function. For that reason, the hybrid protocol proposed applies energy aware mechanisms to the greedy algorithm, in order to gain a better energy performance. The modification to the greedy algorithm is illustrated in Figure 6.5.

When the node x must select a node to forward a packet for the node D, that choice will not consider only the position of the neighbor nodes. Instead, the packet will be forwarded to the node that declares the better energy status, among all the nodes that are nearer to the destination than x. In the figure,

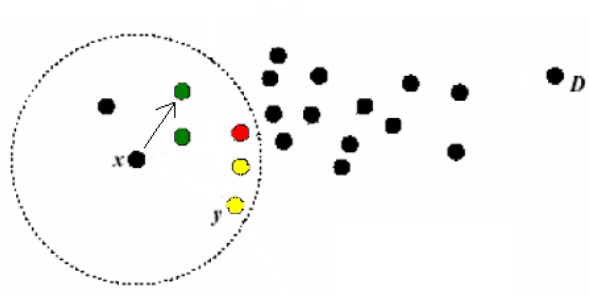

Fig. 6.5. Greedy algorithm modified for nodes energy consideration

all the nodes that are coloured are nearer to the destination, but the source select the node with the higher battery level (and not the nearest to D).

Part III

Evaluation

7

Simulations

In this chapter, all the simulations run to validate the solutions proposed in Chapter 6 are presented. The simulations cover a preliminary comparison of the two main proactive and reactive protocols known in MANETs literature (OLSR and DSR), a presentation of the enhancements of energy-aware OLSR over classical OLSR, a tuning of the MDR's α parameter under different traffic types and a proposal of geographical clustering of the OLSR protocol, using greedy algorithms of GPSR protocol.

The chapter is organized as follows: in Section 7.1 the parameter used in the various simulations are presented, while in Section 7.2 the simulation results are illustrated in details.

7.1 Simulation Parameters

7.1.1 OLSR and DSR comparison

In order to test the energy consumption of mobile nodes under OLSR and DSR protocols and to evaluate the performance of MANET under mobility many simulations were carried out. In particular, the effect of overhearing, idle power, mobility and protocol mechanisms such as route cache reply and link failure notification at data link layer were considered. Route cache reply mechanism was activated in the DSR protocols and the protocol with this mechanism was called `DSR_rc`. On the other hand, the link failure notification at data link layer has been applied to OLSR protocol and the co-respective protocol was called `OLSR_ln`.

To compare the DSR and the OLSR protocols, a dense wireless network was simulated, with 50 nodes moving in a 870 × 870 m^2 area (with a density of about 33 and 66 $nodes/km^2$). Each node moves in this area according to the random waypoint mobility model, with a speed in the range [0, 20] m/s and no pause time.

In terms of traffic, three different situations were studied: in the first case, Idle Power and Overhearing effect have been evaluated (Section 7.2.1).

Then, we considered a fixed connection pattern (FCP), with 12 CBR/UDP sources generating 10 and 20 $packets/s$ (packet size is set to 512 $bytes$) and a variable connection pattern (VCP), where a single connection between two randomly selected nodes (source and destination) of the network is created every 10 s of simulation and lasts 10 s. The duration of each simulation is 450 s, with a startup period in the first 100 s (where no traffic is generated). This means that if each connection lasts 10 s, the first connection starts at 100 s and each connection is generated after the end of the previous connection. Both VCP and FCP were carried out under the second simulation campaigns (Section 7.2.1).

The third case is associated with the node mobility (Section 7.2.1). Different mobility speeds (0, 5, 10, 15 and 20 m/s) were considered and the effect of mobility on the energy dissipation of both OLSR and DSR were analyzed.

The simulation parameters are presented in the following tables. The first table (Table 7.1) presents the common parameters adopted for the different simulation tests. Tables 7.2-7.4 present, respectively, the simulation parameters adopted in each simulation campaign.

Table 7.1. Common simulation parameters

Parameter	Value
Simulation area	870 m × 870 m
Simulation duration	450 s
Connection type	CBR/UDP
Number of traffic source	12
Data packet size	512 $bytes$
Power for transmission P_{tx}	1.4 W
Power for reception P_{rx}	1.0 W
Routing protocols	DSR, OLSR

7.1.2 Energy Aware OLSR

The Energy Aware OLSR proposal has been simulated in a wide range of scenarios, and simulation results under different circumstances were compared with the classical OLSR proposal. In particular, three simulation campaigns were run.

Firstly, a fixed scenario (Section 7.2.2), with 21 mobile nodes statically placed within a 3×7 grid (as shown in Figure 7.1). In this scenario, a single CBR connection is set, between nodes 7 and 13, in order to demonstrate the behavioral difference between the different approaches.

Table 7.2. Simulation parameters for Idle Power and Overhearing

Parameter	Value
Number of mobile nodes	{25, 50}
Maximum node speed	5 m/s
Connection pattern	FCP
Data packet rate for each connection	20 $pkts/s$
Connection duration	15-400 s
Initial node energy	30.0 J
Idle power	{0.0, 0.2, 0.5, 0.9} W
Energy consumption for overhearing	{Yes, No}
Route cache reply (for DSR only)	Yes
Link layer failure notification (for OLSR only)	No

Table 7.3. Simulation parameters for variable and fixed connection pattern

Parameter	Value
Number of mobile nodes	50
Node speed	5 m/s
Connection pattern	{FCP, VCP}
Data packet rate for each connection	{10, 20} $pkts/s$
Connection duration	100-400 s
Initial node energy	10.0 J
Idle power	0.0 W
Energy consumption for overhearing	Yes
Route cache reply (for DSR only)	{Yes, No}
Link layer failure notification (for OLSR only)	{Yes, No}

Table 7.4. Simulation parameters for mobility scenario

Parameter	Value
Number of mobile nodes	50
Node speed	{0.1, 5, 10, 15, 20} m/s
Connection pattern	{FCP, VCP}
Data packet rate for each connection	20 $pkts/s$
Connection duration	100-400 s
Initial node energy	2.0 J
Idle power	0.0 W
Energy consumption for overhearing	No
Route cache reply (for DSR only)	Yes
Link layer failure notification (for OLSR only)	Yes

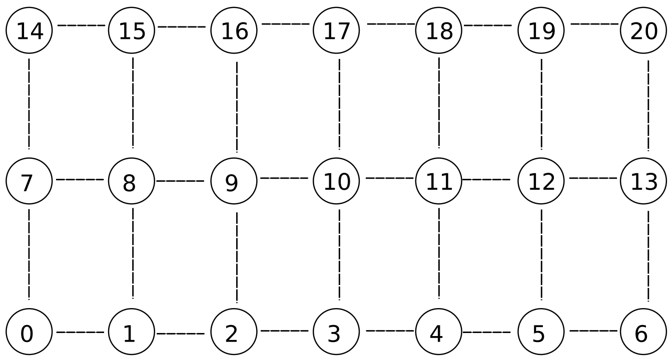

Fig. 7.1. The grid for fixed simulation scenario

Then, a static scenario (Section 7.2.2) was simulated, with 50 nodes randomly placed in an 870 × 870 m^2 area. In this scenario, there were 12 CBR connections between randomly selected different source-destination couples of nodes.

In another simulation campaign (Section 7.2.2), the previous simulations were repeated twice, respectively with a maximum node mobility of 5 and 10 m/s. These simulations validated the performance evaluation of the proposal under mobility.

Finally, the performances of the different simulated protocols with different mobility rates were compared (Section 7.2.2).

Table 7.5 summarizes the parameters used in the static and dynamic simulation campaigns of this work. Each simulation lasts 380 seconds, and the CBR connections start at 30 seconds and end at 350 seconds (giving the protocol an initial interval to exchange routing information and build the routing tables, and a final interval to deliver buffered packets). Transmission and reception power are selected according to the 802.11 network card measurements performed in [16], and the idle power is set to 0 W to neglect its dominating effect over energy consumption, measured in [2].

The simulations of these campaigns aim to compare the classical OLSR protocol with different metrics and mechanisms proposed to increase its energy efficiency. The energy aware metrics simulated were MDR and CMDR. Moreover, the Energy-Aware Willingness setting for OLSR was simulated, both as a standalone mechanism and in conjunction with energy-aware metrics. In the following, all simulated solutions are listed:

- Classical OLSR ("olsr/mtpr")
- Energy-Aware Willingness Setting ("ea-will")

Table 7.5. Simulations parameters for Energy Aware OLSR

Parameter	Value
Modulation	QPSK
Area	870 $m \times$ 870 m
Nodes	50
Nodes speed	0-10 m/s
Simulation Time	380 s
Traffic Sources	12
Traffic Type	CBR/UDP
Packet Size	512 *bytes*
Start of Traffic	30 s
End of Traffic	350 s
Transmission Power	1.4 W
Reception Power	1.0 W
Idle Power	0.0 W

- Minimum Drain Rate Routing ("mdr")
- Conditional MDR Routing ("cmdr")
- MDR+EA-Willingness ("mdr+ea-will")
- CMDR+EA-Willingness ("cmdr+ea-will")

It can be noticed how the classical OLSR approach is assimilated, here, with the MTPR routing metric. In fact, if the transmission power is constant for all the nodes in the network, using a minimum hop metric (as classical OLSR does) corresponds to the use of a Minimum Total Transmission Power Routing, in energy-aware routing terminology.

7.1.3 MDR Tuning

To give a better explanation of our studies, the formulas shown in Section 6.3 were implemented in a spreadsheet, and the actual evaluation of drain rate was simulated at a node with one of consumption models depicted. The evaluation was repeated for different values of the α parameter, so that it could be noticed what value of that parameter best fits to each scenario. The value of drain rate evaluation window (W) was set to 2 seconds and the "on/off" burstiness β to 1/2.

In Table 7.6 there is a summary of the following simulations.

Five scenarios were simulated fot MDR tuning.

In the first scenario (Section 7.2.3), the node has a constant energy consumption, at a fixed drain rate (set to the average drain rate value, as shown in Table 7.6).

Then, an "on/off" consumption model is adopted in the remaining scenarios: the former (Section 7.2.3) shows a period T of 1 second (0.5 seconds consuming at a drain rate double than average, and 0.5 seconds without any

Table 7.6. Simulation parameters for MDR tuning

Parameter	Value
α	{0.1, 0.3, 0.5, 0.7, 0.9}
$T = Ton + Toff$	{0.6, 1.0, 1.1, ∞}
time	up to 150 s
average drain rate	1.0 J/s
β	0.5
W	2 s

consumption), and the others (Sections 7.2.3 and 7.2.3) have a period of 1.1 seconds. It can be immediately noticed how the first period fits perfectly in the evaluation window (exactly, two periods for each window), while the second period does not: in that case, there will be residuals in the calculation of the drain rate, and we can show how it copes with them.

In the fourth scenario (Section 7.2.3), an interval is introduced without energy consumption within the previous case: this allows evaluation of the variable energy consumption patterns.

Finally, the fifth scenario (Section 7.2.3) shows a more realistic case of energy consumption, with sample values not regular in the time, owing to transmission, reception and overhearing of packets ([46], [16], [2]).

7.1.4 Clustered Geographical OLSR

The parameters used for the simulation of the hybrid OLSR-GPSR protocol proposed in Section 6.4 are presented in Table 7.7.

Table 7.7. Simulations parameters for Clustered Geographical OLSR

Parameter	Value
Area	1000 $m \times$ 1000 m
Nodes	{30, 40, 50, 60}
Simulation Time	400 s
Traffic Sources	{6, 10, 14}
Traffic Type	CBR/UDP
Packet Size	512 $bytes$
Send rate	6 $pkts/sec$
Start of Traffic	30 s
End of Traffic	380 s
Transmission Power	1.4 W
Reception Power	1.0 W
Idle Power	0.0 W

7.2 Simulation Results

7.2.1 OLSR and DSR comparison

Idle Power and Overhearing influence

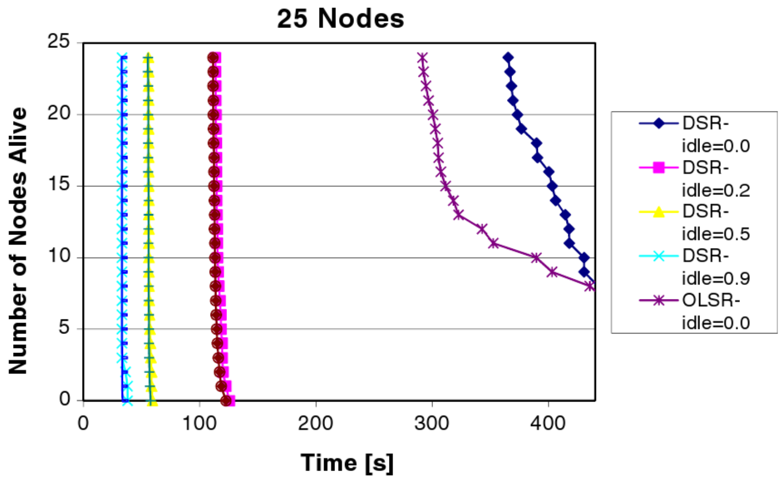

Fig. 7.2. Number of alive nodes vs time varying idle power with N = 25

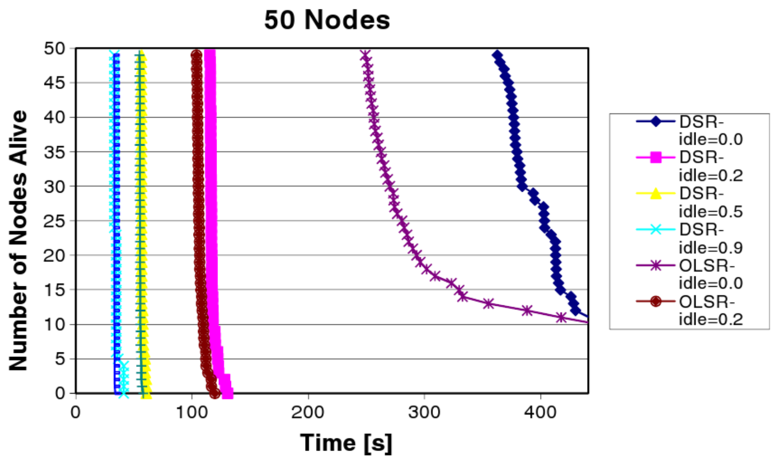

Fig. 7.3. Number of alive nodes vs time varying idle power with N = 50

The first task is to evaluate the influence of Idle Power and Overhearing over energy consumption in a MANET. These effects reduce the network lifetime, consuming rapidly the nodes' batteries with very low differences between reactive and proactive protocols. As we can notice from Figures 7.2 and 7.3, even with a low idle state energy consumption, all the nodes in the network tend to exhaust their battery at the same time (i.e., when idle power consumes all the device energy), no matter if one is evaluating DSR or OLSR protocol. If the ideal case of no power consumption in Idle mode is considered and just the overhearing effect is accounted for, nodes can live longer for both DSR and OLSR. It is possible to observe that nodes under OLSR die earlier than nodes under DSR.

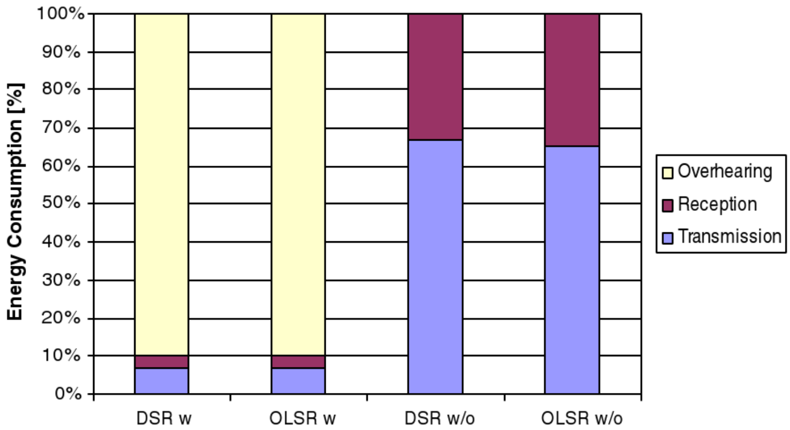

Fig. 7.4. Energy percentage consumption by type with (w) and without (w/o) overhearing effect for 25 nodes

The DSR and OLSR protocols were simulated with and without theoverhearing effect. The results, from the energetic point of view, can be seen in Figures 7.4 and 7.4. When node density increases the overhearing effect is predominant and a lot of energy is dissipated (around 90-95%). However, when the overhearing dissipation is not accounted for, power dissipations in transmission and reception phase maintain the same proportion and they are not affected by node density increases.

Since Idle Power and Overhearing effects dominate the energy consumption in the simulation of a dense, high-traffic loaded network, to evaluate the actual differences between reactive and proactive protocols in a MANET from energetic point of view, both of these effects will be ignored in the rest of the simulations. In the implementation of DSR protocol, this last consideration leads to an important remark. When the energy consumption in overhearing

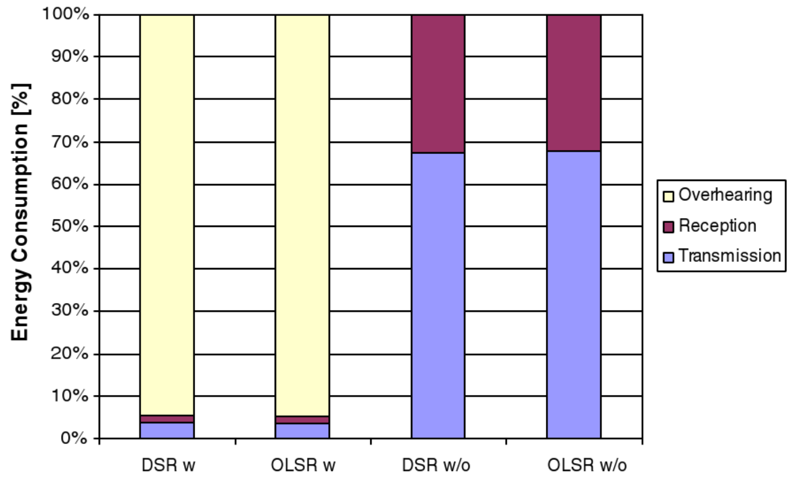

Fig. 7.5. Energy percentage consumption by type with (w) and without (w/o) overhearing effect for 50 nodes

packets is neglected, the promiscuous mode of the protocol must be turned off. This means that DSR cannot rescue routing information from packets directed to another node. Therefore, in the rest of this simulation, the DSR protocol will be considered without the promiscuous mode operation.

Fixed and variable connection pattern

In these simulation campaigns the constant and variable traffic load over the MANET were considered. It was decided to adopt two connection patterns because they stress the network in different ways. In particular, variable connection pattern (VCP) forces DSR to start more route discovery procedures while static connection pattern (Fixed Connection Pattern) stresses OLSR that sends a lot of control packets together with data packets, quickly exhausting the energy.

Fixed connection patter (FCP)

In this first case, the network experiments a high, static traffic load, with 12 CBR/UDP traffic sources sending a constant amount of data between 100 and 400 simulation seconds and two data rates of 10 and 20 packets per seconds ($pkts/s$) are considered. Figures 7.6 and 7.7 show the percentage of remaining nodes in the network over time, plotting the halt-time of mobile nodes.

The more the curve is on the top right of the plot, the more the protocol prolongs the nodes' lifetime (thus prolonging the lifetime of the entire network). It can be seen how DSR takes advantage of its reactive nature: in the first 100 seconds of simulation, while OLSR spends energy to update the

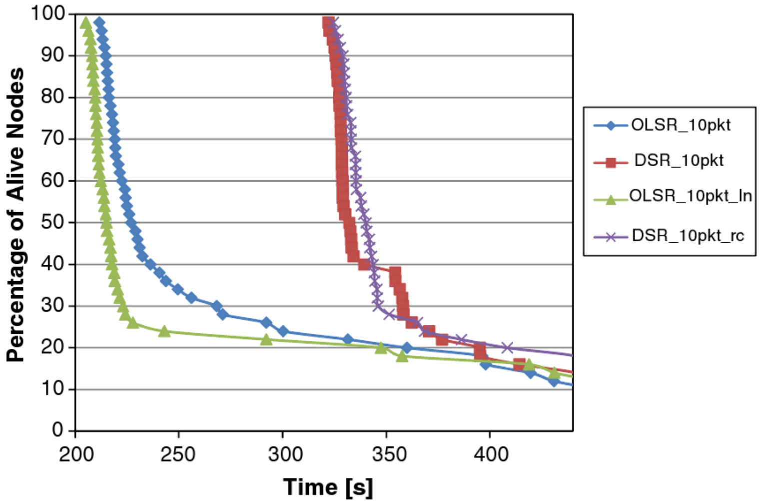

Fig. 7.6. Alive nodes vs time with fixed connection pattern

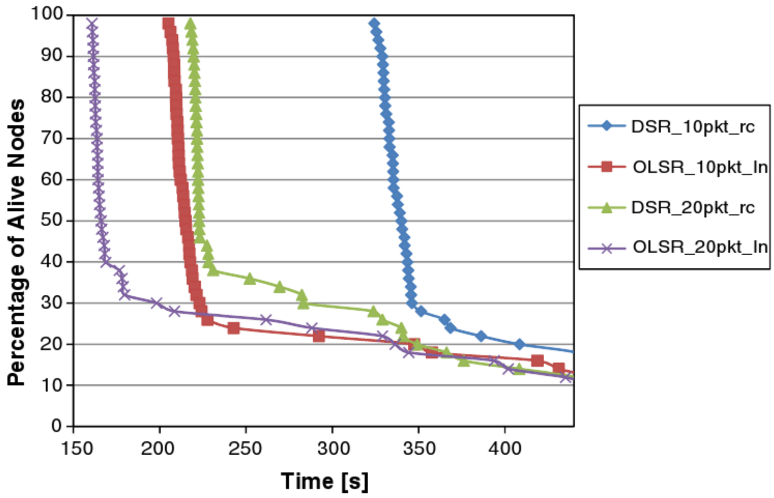

Fig. 7.7. Alive nodes vs time with fixed connection pattern and different data packets rate (10 and 20 pkts/s)

network topology, DSR does not generate packets (because there is no data transmission in the network). However, the gap is between 30 and 80 s for data rate of 20 $pkts/s$, showing the good performances of OLSR with high traffic rates. However, when data traffic rate is lower (10 $pkts/s$) the gap between OLSR and DSR is greater (about 140 s). To have a better vision of the behavior of the routing protocols with respect to the traffic, the lifetime of the connections of the simulated MANET can be plotted. Figure 7.8 shows how the response of OLSR and DSR is very similar (but, obviously, shifted: the proactive protocol starts its periodic exchange of message at the beginning of the simulation). An increase in the data traffic rate produces a reduction in the connection lifetime.

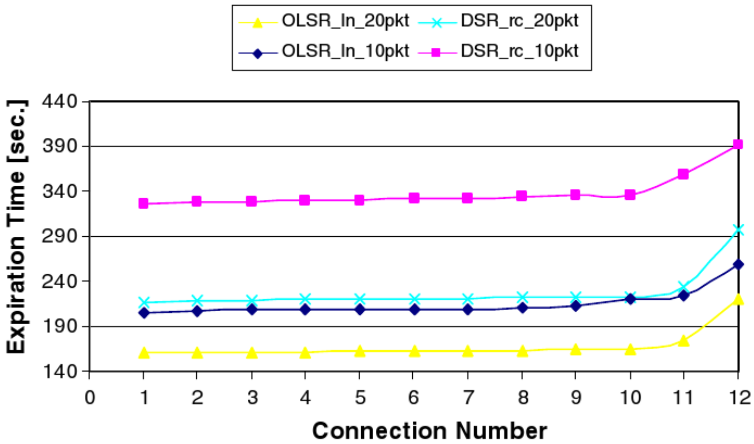

Fig. 7.8. Connection expiration time with a fixed connection pattern and different data packet rate (pkts/s)

Other parameters are shown in Tables 7.8 and 7.9. DSR and `DSR_rc` present higher E2E delay in comparison with OLSR. This is due to the reactive nature of DSR that determines a high number of Route request to find a new path from source to destination. On the other hand, OLSR presents very low E2E delay due to the proactive info management that permits to have the path immediately available.

However, DSR offers a higher DPR because it saves more energy in the fixed Connection Pattern scenario. OLSR, on the contrary, drains the energy faster producing the death of more nodes, such as shown in the previous graphics, and causing the network partition. Concerning the route cache reply for DSR (`DSR_rc`) and link failure notification at data link layer (`OLSR_ln`), it is possible to see an improvement, respectively, of Overhead for DSR and of Packet Delivery for OLSR. Moreover, high traffic load (20 $pkts/s$) determines a reduction of DPR and an increase in Overhead for both DSR and OLSR.

It can be seen how the overhead of OLSR is considerably higher than the one of DSR. The data packet delivery ratio is very different between the two protocols. To know the reason, the throughput of the dynamic scenario over simulation time was plotted in Figures 7.9 and 7.10.

In Figure 7.9, the data throughput of OLSR is lower than DSR, because OLSR wastes more bandwidth for control overhead (O/H) and it is not able to

Table 7.8. DSR vs OLSR performance evaluation with a data packet rate of 10 pkts/s and FCP

	Packet delivered	Overhead	E2E delay (ms)	Alive nodes (%)
DSR	29658.42	12.25	14.10	14
OLSR	9755.519	39.24	6.07	12
DSR_ rc	30354.55	12.37	28.21	20
OLSR_ln	14402.58	25.89	5.53	18

Table 7.9. DSR vs OLSR performance evaluation with a data packet rate of 20 pkts/s and FCP

	Packet delivered	Overhead	E2E delay (ms)	Alive nodes (%)
DSR	27460.98	18.87	13.03	10
OLSR	10395.12	47.24	17.66	8.8
DSR_rc	28193.32	18.47	36.32	16
OLSR_ln	12556.57	36.23	11.48	15

adapt itself faster to topological change due to mobility (5 m/s). However, if the data link notification (`OLSR_ln`) is applied, the data throughput increases a lot and performance similar to DSR is obtained (a data throughput of 120,000 *bytes* for both DSR and `OLSR_ln`). It is possible to see also the reduction in the duration of high throughput due to the faster node energy consumption that led to node death and network partitioning.

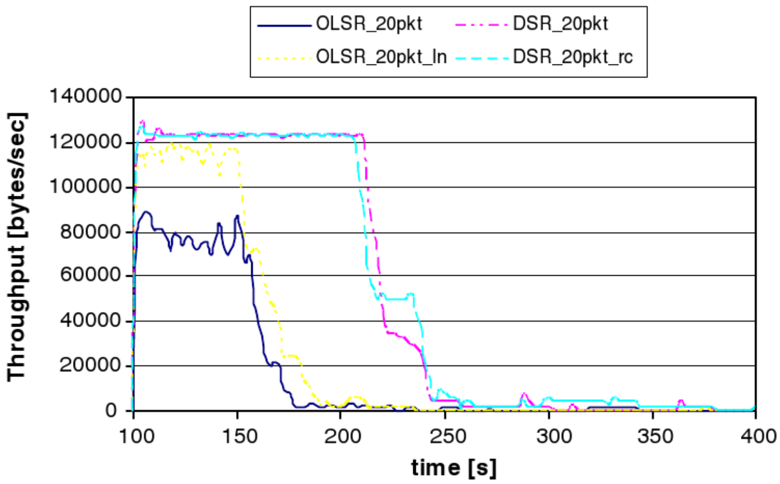

Fig. 7.9. Throughput vs time with fixed connection pattern

Moreover, in Figure 7.9, it is possible to see as the throughput values are coherent with the percentage of alive nodes. When around 150 s a lot of nodes

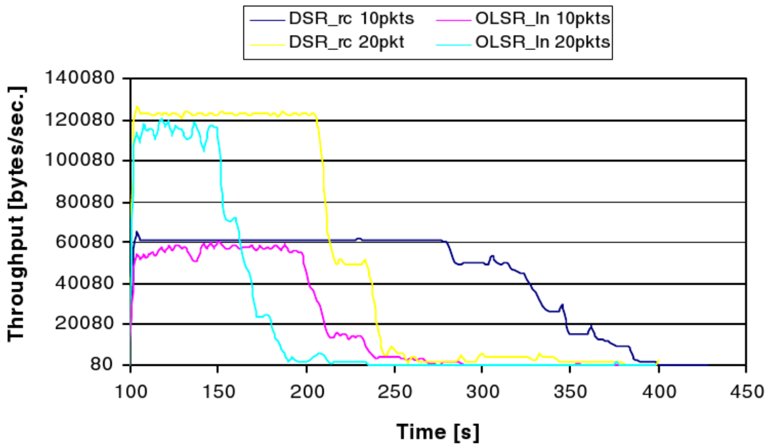

Fig. 7.10. Throughput vs time with fixed connection pattern and data packet rate of 10 and 20 pkts/s

die under the OLSR protocol, the data throughout decreases. In the same way, the throughput of DSR is reduced around 250 s when the greater number of nodes die (40 nodes).

When data rate is reduced, a stable throughput can be supported by both OLSR and DSR for a longer time in comparison to 20 $pkts/s$ of data rate. This is due to the energy saving in the transmission and receiving power of mobile nodes. Also in this case DSR outperforms the OLSR in terms of longer duration of data throughput.

Before the expiration of connections, DSR presents a stable throughput, while the one of OLSR varies a lot. This is because DSR, being reactive, rapidly reacts to path changes, while these changes lead to packet losses in OLSR. This could be repaired updating the routing tables of OLSR more frequently, but this could lead to very high values of routing overhead.

Variable connection pattern (VCP)

In a second phase, the same, dynamic network topology was simulated to have a variable connection pattern: in this case, a random connection (512 *bytes* packets, sent at a rate of 10 and 20 $pkts/s$) is generated every 10 simulation seconds. Every connection lasts exactly 100 s. In this scenario, the reactive protocol will have to work a little more, to continuously find new routes to the destinations added by the connection pattern. Figures 7.11 and 7.12 show nodes' lifetime, for the simulated network.

DSR without cache reply and with cache reply (`DSR_rc`) presents similar performance in terms of node lifetime. When data rate decreases (10 $pkts/s$) the node lifetime increases by 100 s. When simulation reaches 300 s, 40 nodes

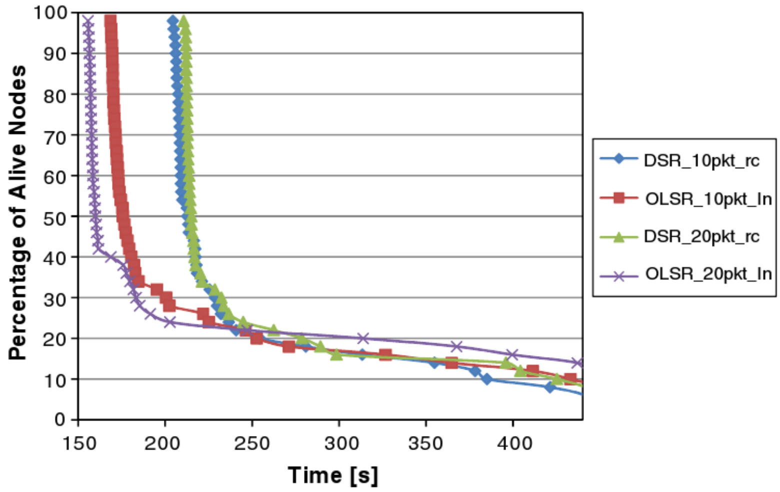

Fig. 7.11. Alive nodes vs time with variable connection pattern

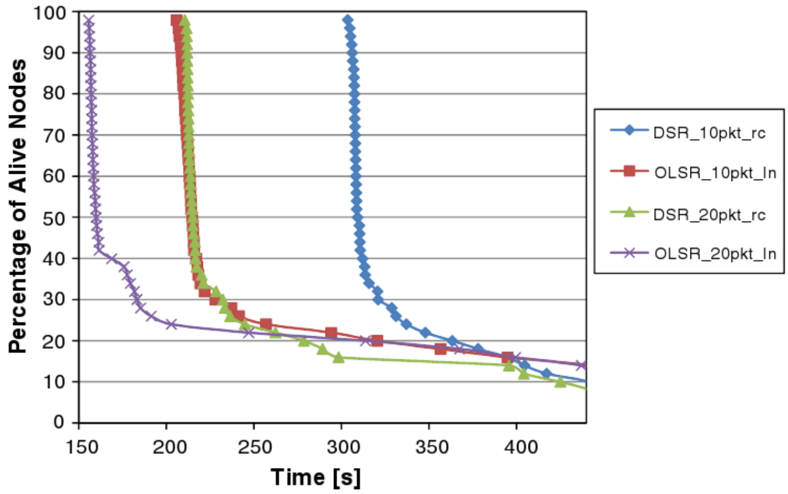

Fig. 7.12. Alive node vs time with variable connection pattern and different data packet rate (10 and 20 pkts/s)

die for both DSR and OLSR. OLSR consumes more energy than DSR and this determines a shorter node lifetime such as shown in Figures 7.11 and 7.12 where, for 150-180 s, 50% of nodes die. When data rate decreases (10 $pkts/s$) the lifetime of nodes increases of 80 s under OLSR protocol. It is possible also to see as there is no difference in terms of number of alive nodes if we apply route cache reply in DSR or not. This behavior will be explained in the following through results in Tables 7.10 and 7.11. Moreover OLSR for high data rate (20 $pkts/s$) show a higher number of alive nodes after 270 s. Also this behavior is attributed to the cache reply use of DSR that determines a slightly higher energy consumption. For heavy traffic load (20 $pkts/s$) also DSR degrades its performance and a lot of nodes die (more than 80% after 260 s). This is due to the DSR activity that causes many simultaneous lacks of energy, corresponding to RREQ broadcast storms.

Table 7.10. DSR vs OLSR performance evaluation with a data packet rate of 10 pkts/s and VCP

	Packet delivered	E2E delay (ms)	Overhead	Alive nodes (%)
DSR	25840.83	41.14	3.64	7
OLSR	10481.31	27.36	46.16	10
DSR_rc	25393.49	62.19	3.70	9
OLSR_ln	12975.73	11.01	34.50	13.6

Table 7.11. DSR vs OLSR performance evaluation with a data packet rate of 20 pkts/s and VCP

	Packet delivered	E2E delay (ms)	Overhead	Alive nodes (%)
DSR	27182.65	81.35	2.22	6
OLSR	13146.21	7.78	27.19	9
DSR_rc	28459.62	77.53	1.87	10
OLSR_ln	14619.54	4.51	23.81	14

In Tables 7.10 and 7.11, DPR, E2E Delay and protocol control overhead (O/H) are listed for all protocols and for both data rates (10 and 20 $pkts/s$). DSR delivers more data packets than OLSR because its lower energy consumption determines a longer node lifetime in comparison to mobile nodes under the OLSR protocol. Concerning the mechanisms of OLSR (link layer notification) and route cache reply, it is possible to observe an improvement in the DPR. Also O/H is reduced especially for lower data rate (10 $pkts/s$) and for OLSR with link layer notification (`OLSR_ln`).

To better justify the low value of OLSR data packets delivery ratio, we plotted the throughput over time for this simulation in Figure 7.13.

As in the previous case (fixed connection pattern), the DSR throughput over time shows an almost stable behavior, while OLSR value changes frequently with time. A light stabilization in the data throughput of OLSR is observed in the case of lower traffic load (10 $pkts/s$) and link layer notification (`OLSR_ln`) (Figure 7.14).

If the route cache reply in DSR is considered, a light data throughput improvement is observed. Moreover, DSR and `DSR_rc` present a more stable throughput and this is due to a greater capacity of DSR to react to link

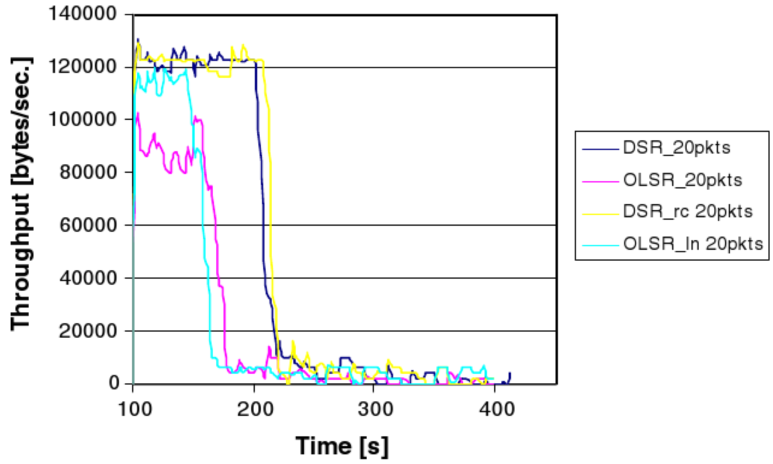

Fig. 7.13. Throughput vs time with variable connection pattern

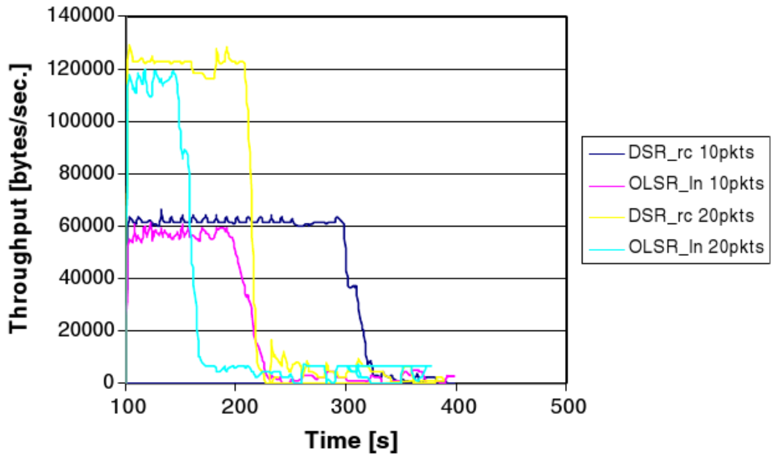

Fig. 7.14. Throughput vs time with variable connection pattern and different data rates (10 and 20 pkts/s)

breakage caused by node mobility (5 m/s). It is interesting to observe also the improvement of OLSR when link layer notification is adopted (`OLSR_ln`) such as testified also by increasing in the DPR (see Tables 7.10 and 7.11). In this case, `OLSR_ln` is able to adapt faster to topology change and to offer a throughout comparable with DSR. However, the duration of high throughput of OLSR is shorter than DSR and `DSR_rc` because more energy is consumed and more nodes die reducing the network connectivity.

Influence of mobility over performance

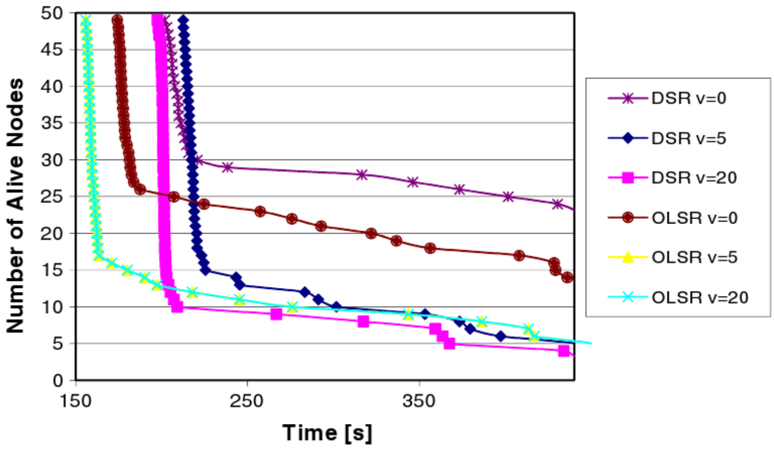

Fig. 7.15. Alive nodes vs time with fixed connection pattern and different nodes mobility (v = 0, 5 and 20 m/s)

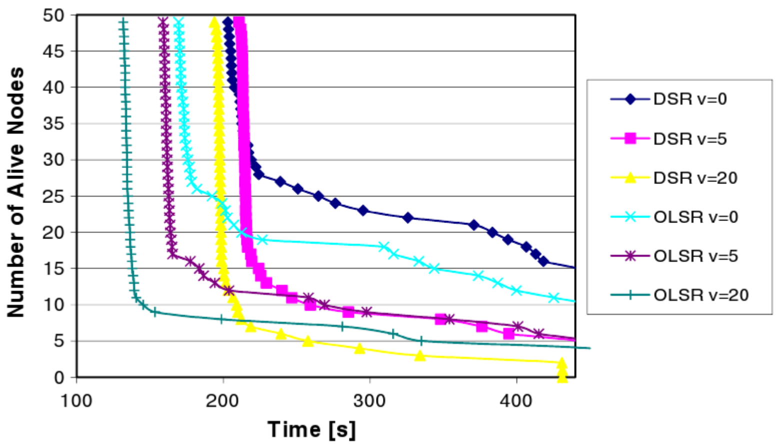

Fig. 7.16. Alive nodes vs time with variable connection pattern and different nodes mobility (v = 0, 5 and 20 m/s)

Figures 7.15 and 7.16 present the number of live nodes under mobility scenarios where $v = 0$, 5 and 20 m/s have been considered. It is possible to observe the good performance of OLSR when no nodes mobility is considered. This is due to the reduction in TC packets sent on the network that allows a longer node lifetime. DSR outperforms OLSR in terms of energy consumption during the simulation because more nodes under DSR are alive. For node mobility of 5 and 20 m/s also DSR degrades its performance because 75% of nodes die in the first 220 s for $v = 20$ m/s and 80% of nodes die in the first 200 s. For a speed of 20 m/s DSR and OLSR consume similar energy and the number of nodes alive is the same. This is due to the high node mobility that forces DSR to start more route discovery procedure consuming energy resources and reducing the benefits of the reactive data management approach.

To better see the slow reaction of OLSR to path changes in the network, the throughput with time with different nodes speeds was plotted, in Figures 7.17 and 7.18. OLSR throughput is maintained for a shorter time than DSR throughput. This is due to higher energy dissipation and nodes death that reduces the network connectivity. Both DSR and OLSR decrease throughput maintenance time for increasing nodes speed because higher speeds imply higher O/H and greater energy dissipation. When node mobility is high (20 m/s) data throughput is more variable for the slower topology change adaptation of OLSR in comparison with DSR.

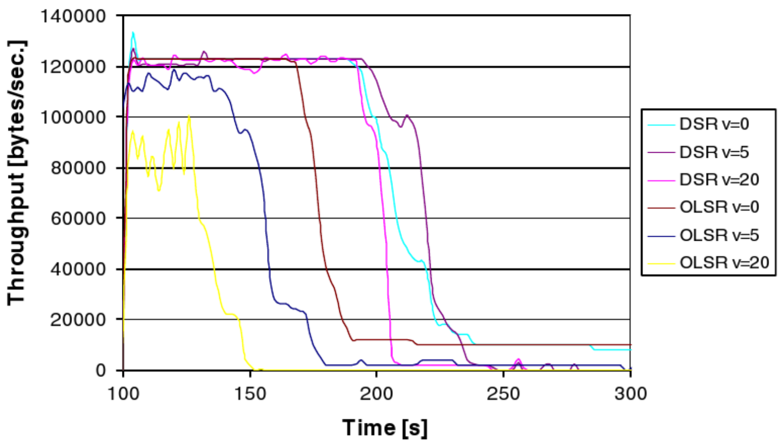

Fig. 7.17. Data throughput vs time with fixed connection pattern and different nodes mobility (v = 0, 5 and 20 m/s)

From the figures above, it is clear how DSR rapidly reacts to topology changes, while OLSR cannot reach the same performance values. DSR in-

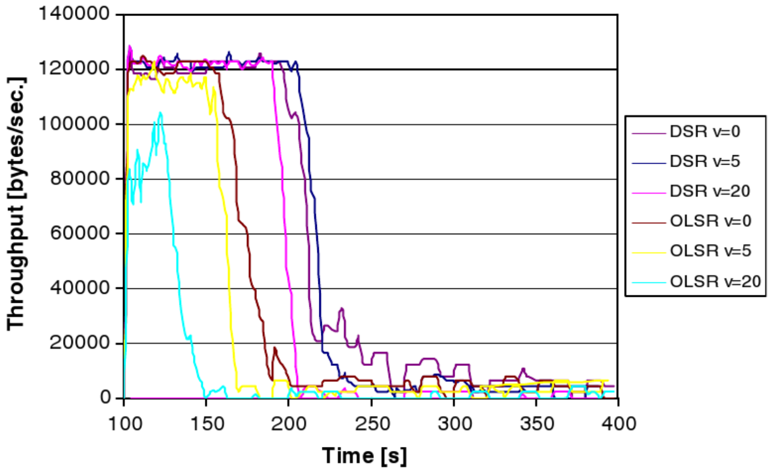

Fig. 7.18. Data throughput vs time with variable connection pattern and different nodes mobility (v = 0, 5 and 20 m/s)

creases its O/H for higher nodes speed such as OLSR as shown in Tables 7.12 and 7.13. However, DPR of OLSR significantly decreases when node mobility is considered because a high data throughput is supported for a shorter time in comparison with the throughout supported by DSR. Moreover, a lot of nodes die and some network partition occurs reducing the DPR. However, in terms of E2E delay, OLSR performs better than DSR and for high mobility (1520 m/s) there is a reduction of 15-20 ms in comparison with DSR. The proactive data management gives the possibility of immediately having a path towards destination. In the DSR protocol a greater route discovery latency determines higher E2E data packet delay.

Table 7.12. OLSR Performance evaluation under different node mobility

m/s	Overhead	E2E delay (ms)	DPR
0	15.61	3.82	24466.15
5	27.31	5.97	13962.43
10	34.53	5.78	10871.79
15	57.14	10.45	7586.69
20	68.71	13.68	6279.92

Table 7.13. DSR Performance evaluation under different node mobility

m/s	Overhead	E2E delay (ms)	DPR
0	0.84	6.76	29561.22
5	1.59	10.91	28031.21
10	2.61	23.42	25679.15
15	3.25	20.07	24361.45
20	3.72	30.32	24392.46

7.2.2 Energy Aware OLSR

Fixed Scenario

Figure 7.19 shows the number of nodes alive during simulation, with different protocols: it clearly demonstrates the effectiveness of energy-aware metrics in increasing nodes lifetime, by means of distributing the energy consumption among nodes not in the minimum-hop path to destination. OLSR/MTPR, not considering the energy status of nodes, cannot select different paths to deliver data, and this leads to higher energy consumption for nodes on the critical path.

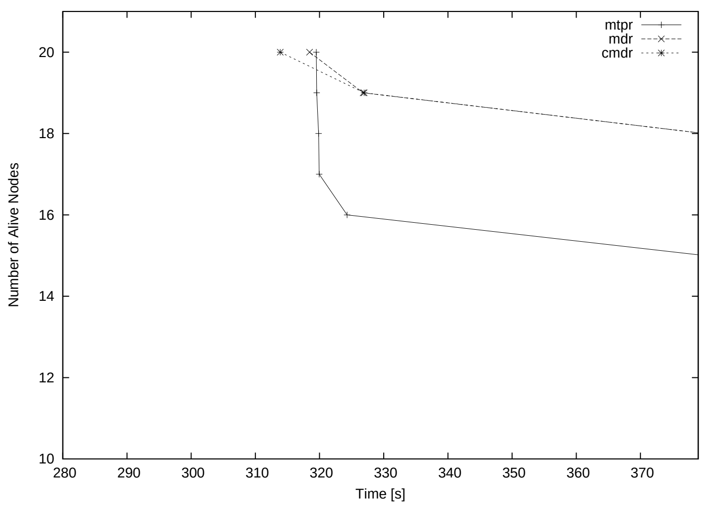

Fig. 7.19. Number of alive nodes over time, in fixed scenario

Figures 7.20 and 7.21 illustrate the selection of path in two different approaches (OLSR and MDR+EA-Willingness): in the former, nodes on the minimum path between source and destination consumed all their energy, while other nodes were not affected by energy consumption; on the other hand, in the latter the energy consumption is distributed among a larger number of nodes, thus preserving the battery of nodes on the critical path to destination.

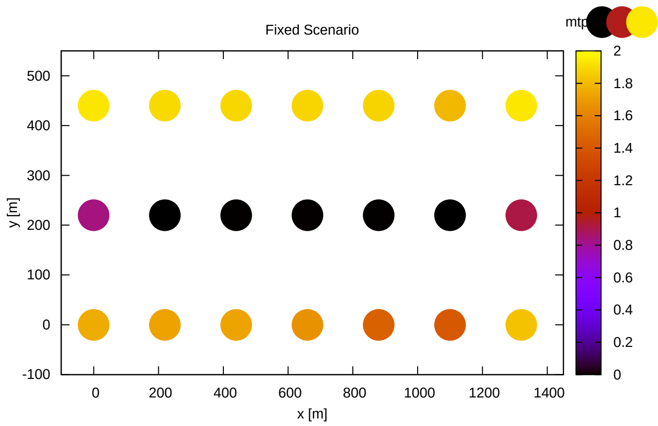

Fig. 7.20. Spatial distribution of Residual energy, with OLSR protocol in fixed scenario

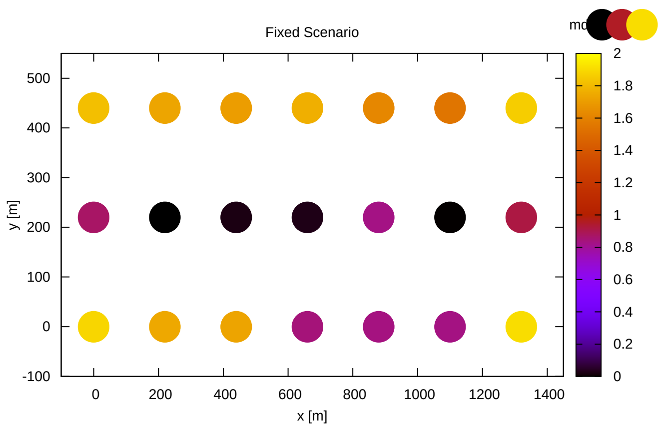

Fig. 7.21. Spatial distribution of Residual energy, with MDR metric and EA-Willingness, in fixed scenario

The difference in energy consumption between MTPR and MDR approaches is also shown in Figure 7.22, where the number of nodes with a high, medium and low energy level at the end of the simulation are counted. It can be noticed how MTPR leaves a larger number of nodes in a good energy status, together with a larger number of nodes in a bad energy status. On the other hand, MDR performs a better distribution of energy consumption, thanks to its energy awareness in path selection.

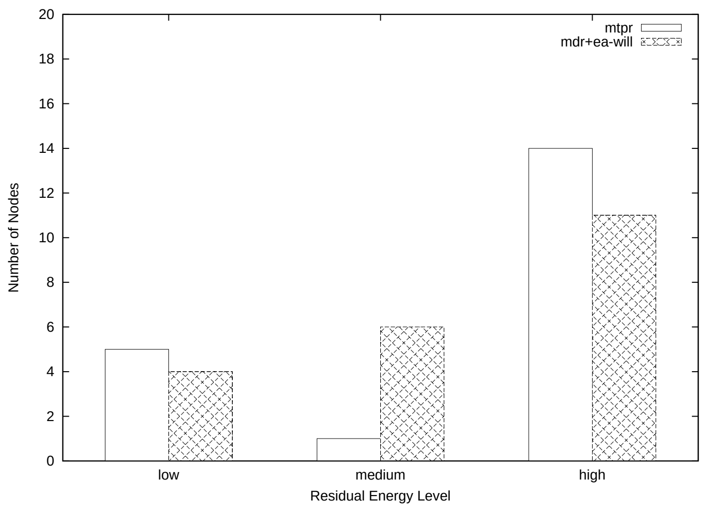

Fig. 7.22. Number of nodes with different residual energy levels, in fixed scenario

Static Scenario

The static scenario simulations show how energy-aware metrics, especially if used in conjunction with the energy-aware willingness setting, can lead to a better energy performance of the network. For example, Figure 7.23 illustrates the number of alive nodes over time with different solutions: the lifetime of nodes increases significantly using MDR metric and EA-Willingness mechanism.

The same behavior is represented in Figure 7.24, where the connection duration is shown. In this case, too, it can be noticed how the use of MDR with EA-Willingness prolongs the duration of most connections.

The measure of average node energy over time, shown in Figure 7.25, illustrates the drawback of energy aware metrics with respect to minimum-hop routing metrics: the distribution of routing among a larger number of nodes, while preserving the energy of the most stressed ones, leads to a higher overall energy consumption in the network. Under this aspect, the better

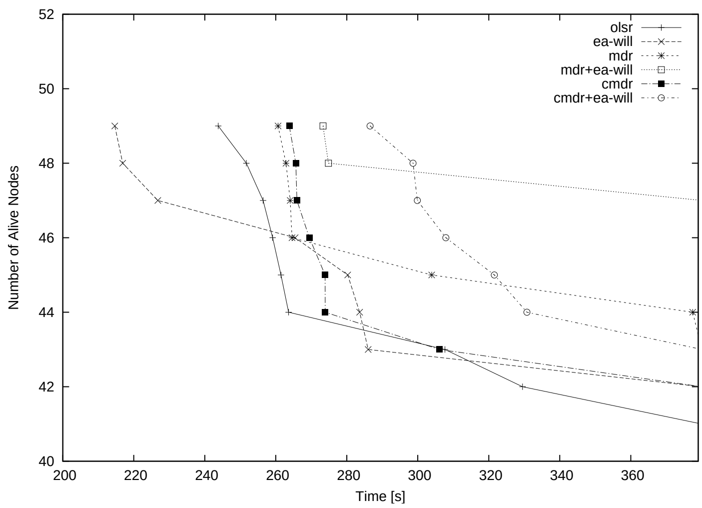

Fig. 7.23. Number of alive nodes over time, in static scenario

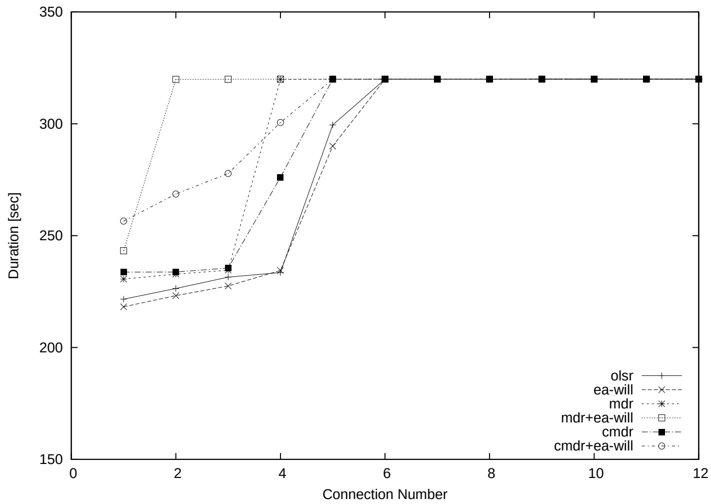

Fig. 7.24. Connections duration, in static scenario

energy-aware metric is CMDR, that improves average energy consumption behavior by the adoption of a hybrid approach between MTPR and MDR metrics.

In order to evaluate the influence of different routing approaches on the selection of MPR in OLSR protocol, Figure 7.26 shows the number of MPR per node over time with different protocols. Energy-aware metrics, as shown, tend to increase the number of MPR of the network, aiming to distribute energy consumption over a larger set of nodes.

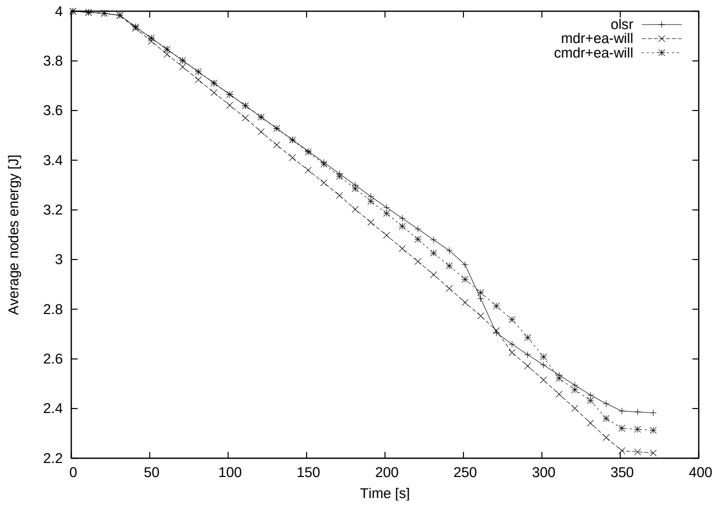

Fig. 7.25. Average nodes energy, in static scenario

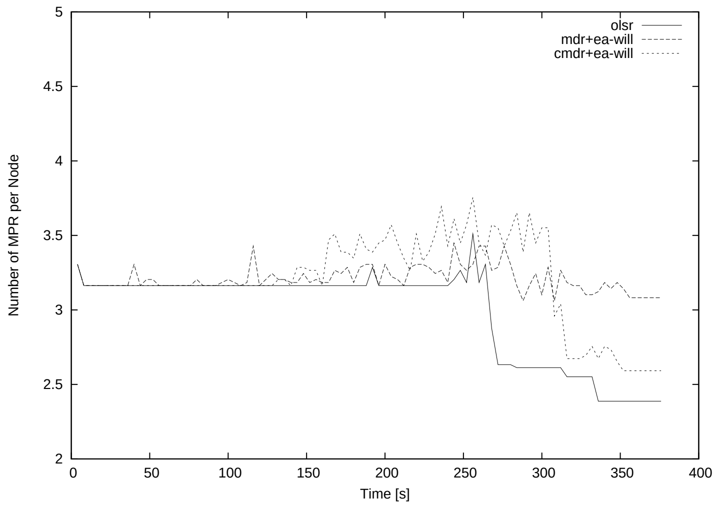

Fig. 7.26. Number of MPR per node, in static scenario

Figure 7.27 illustrates the advantage of using energy-aware metrics in an energy-constrained network in terms of aggregate throughput. The improvement of the throughput led by MDR and CMDR with EA-Willingness, with respect to the classical OLSR protocol, is clearly shown here.

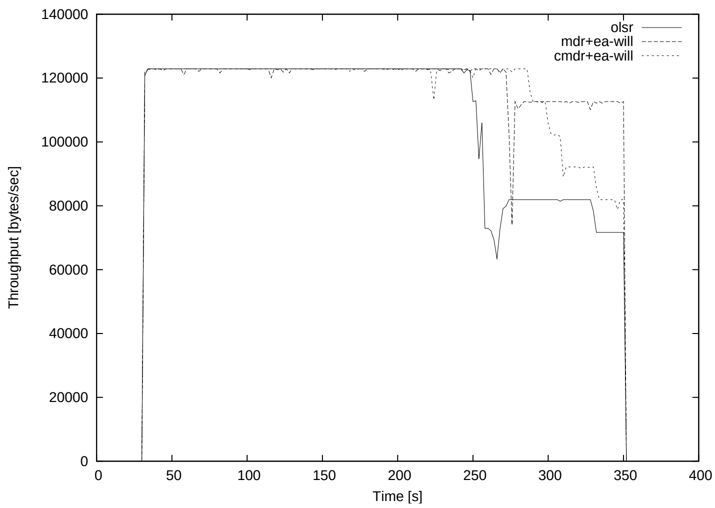

Fig. 7.27. Aggregate throughput over time, in static scenario

Dynamic Scenarios

Introducing mobility of nodes in the simulated scenario, the same behavior of the analised protocols can be observed. Figures 7.28-7.32 illustrate the results of simulations with a maximum node speed of 5 m/s. The first figure shows the lifetime of nodes in the network: it can be noticed how energy-aware routing metrics, especially MDR used in conjunction with EA-Willingness, effectively prolong the life of first nodes to end their battery capacity, while the lifetime of other nodes results longer using classical OLSR protocol. This fact confirms the hypothesis about the distribution of energy consumption among nodes: while OLSR tends to consume the battery of a subset of nodes in the network, energy-aware metrics succeed in distributing such a consumption over a larger number of nodes.

The advantage of using energy-aware metrics in energy constrained networks appears in Figure 7.29, where the connection duration is depicted. In fact, energy-aware metrics obtain longer lasting connections compared to classical OLSR protocol, and the use of EA-Willingness selection leads to even better results.

The average energy with mobility of 5 m/s is illustrated by Figure 7.30. As noticed in previous simulations, the use of MDR metric leads to a higher overall energy consumption in the network, while the performance of CMDR remains close to MTPR metric (used by classical OLSR).

The results change only if the average energy of the subset of MPR nodes in the network is considered, as in Figure 7.31. Here it can be noticed how, when the energy begins to lack, energy-aware metrics start to perform better

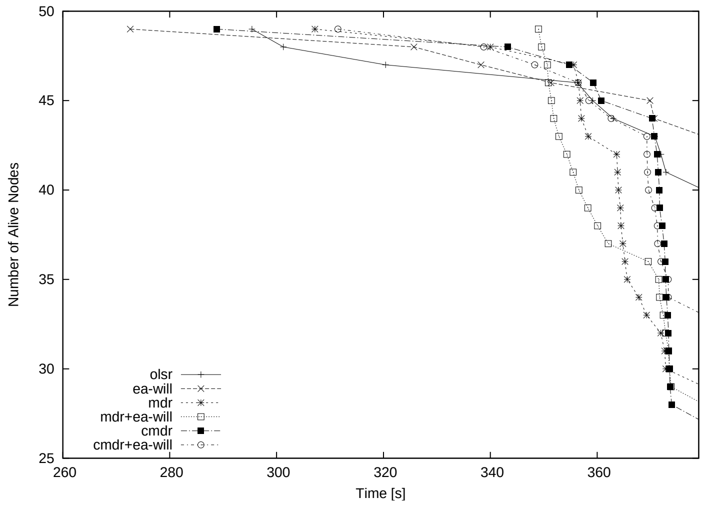

Fig. 7.28. Number of alive nodes over time, at 5 m/s

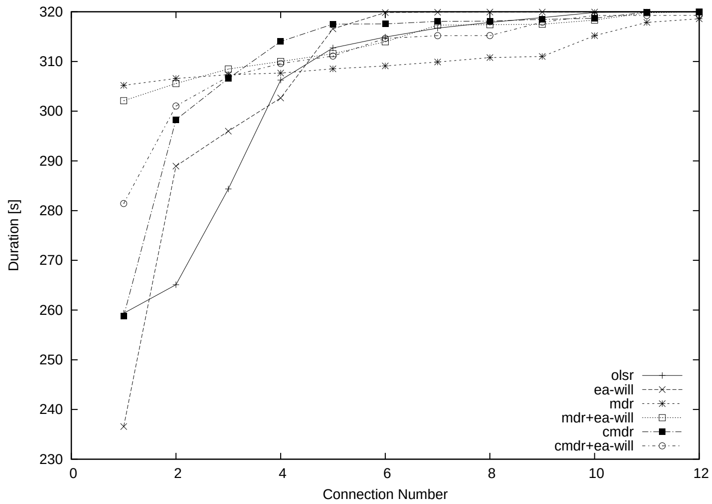

Fig. 7.29. Connections duration, at 5 m/s

than OLSR: these protocols, in fact, tend to select as MPR the nodes with a better energy profile in the network, thus prolonging connection lifetime.

In terms of aggregate throughput, the situation at 5 m/s is illustrated in Figure 7.32: energy-aware metrics lead to higher values. Oscillations in this figure are due to node mobility, that causes continuous break of links, with consequent packet losses.

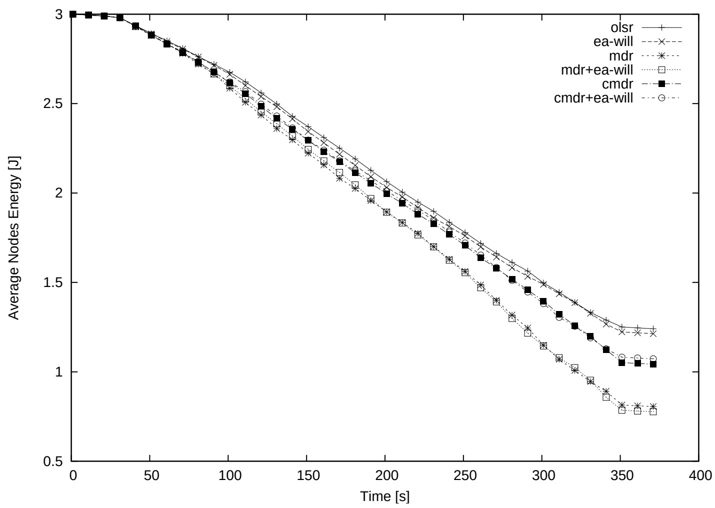

Fig. 7.30. Number of alive nodes over time, at 5 m/s

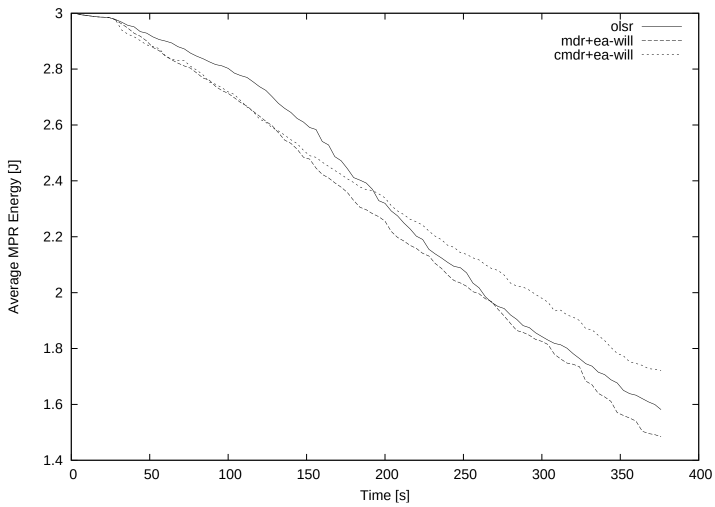

Fig. 7.31. Average MPR energy, at 5 m/s

In Figures 7.33-7.38, the results of simulations with node mobility of 10 m/s are presented. In the first one, showing the lifetime of nodes, the same behavior of previous cases can be observed: OLSR causes the earliest lack of energy, MDR causes a lower number of nodes to reach the end of simulation, and CMDR unites the benefits of the two approaches.

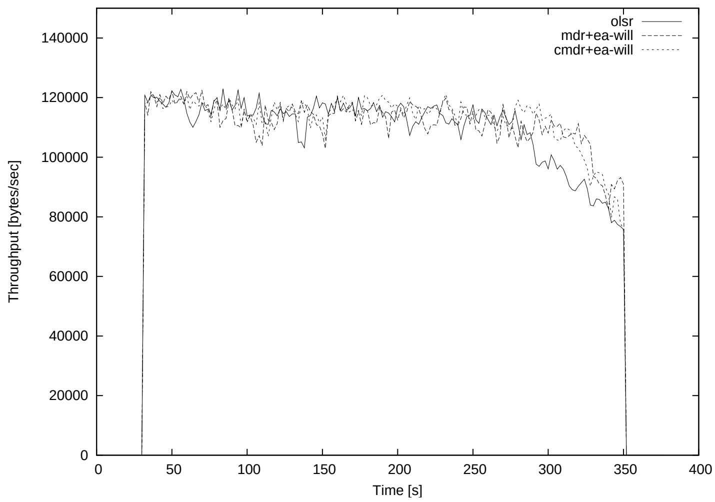

Fig. 7.32. Aggregate throughput, at 5 m/s

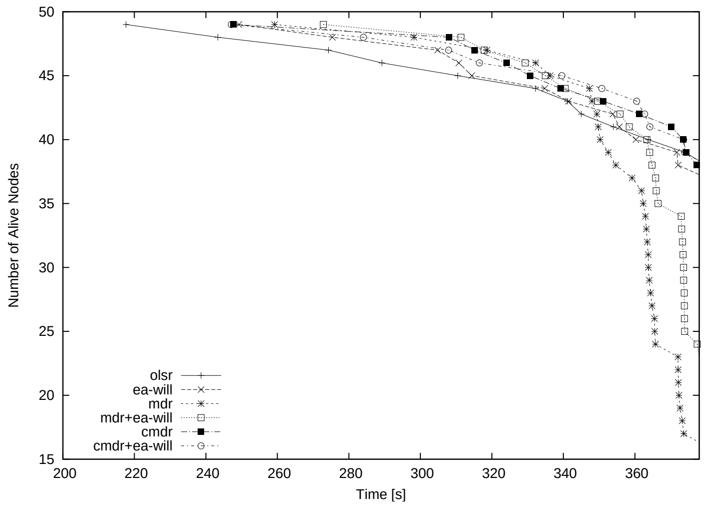

Fig. 7.33. Number of alive nodes over time, at 10 m/s

Figure 7.34 demonstrates how energy-aware metrics, even in the presence of high nodes mobility, result in longer connection durations (especially MDR protocol with EA-Willingness).

In Figure 7.35 it can be noticed how the behavior of energy consumption in the network is very similar between OLSR and CMDR, at 10 m/s, while MDR keeps consuming more energy, on average.

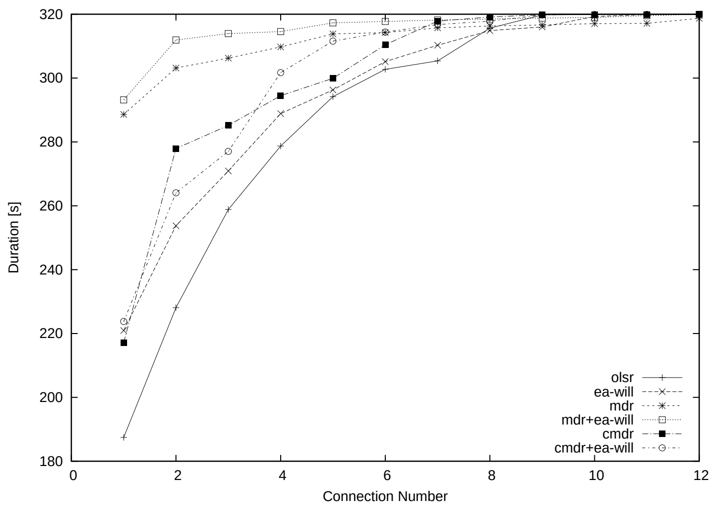

Fig. 7.34. Connections duration, at 10 m/s

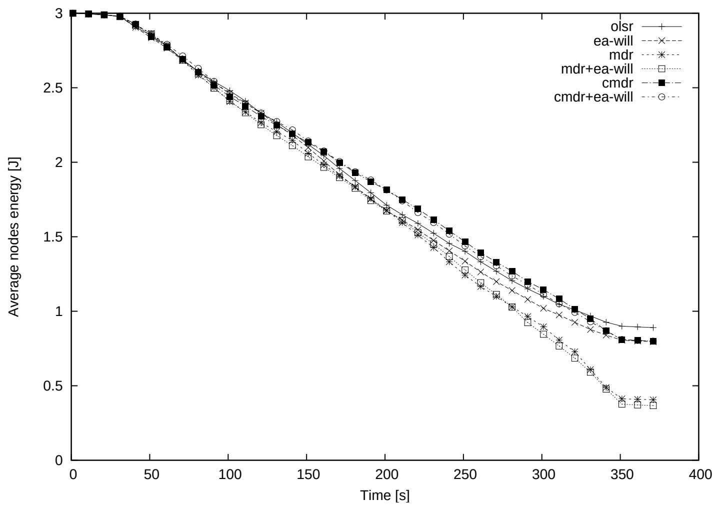

Fig. 7.35. Average nodes energy over time, at 10 m/s

The same behavior can be observed plotting the average energy only for MPR nodes, as in Figure 7.36: OLSR and CMDR obtain better results, but MDR protocol still remains close to their average consumption values.

Figure 7.37 illustrates the number of MPR per node over time: especially in the last part of the simulation, only energy-aware metrics succeed in maintaining a large number of MPR.

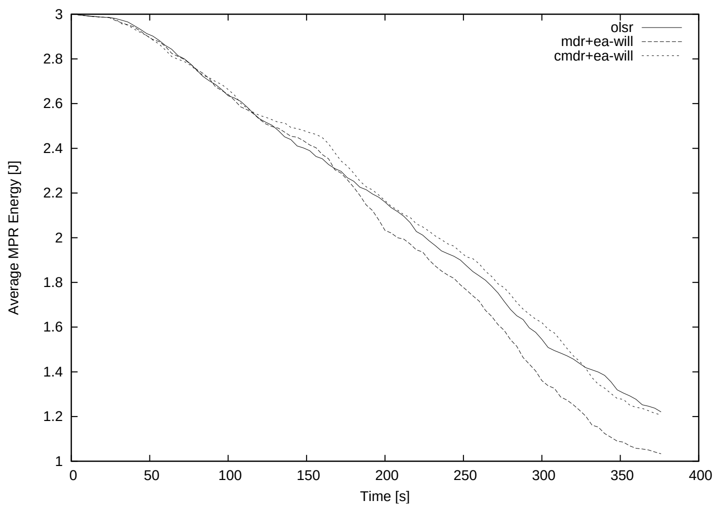

Fig. 7.36. Average MPR energy over time, at 10 m/s

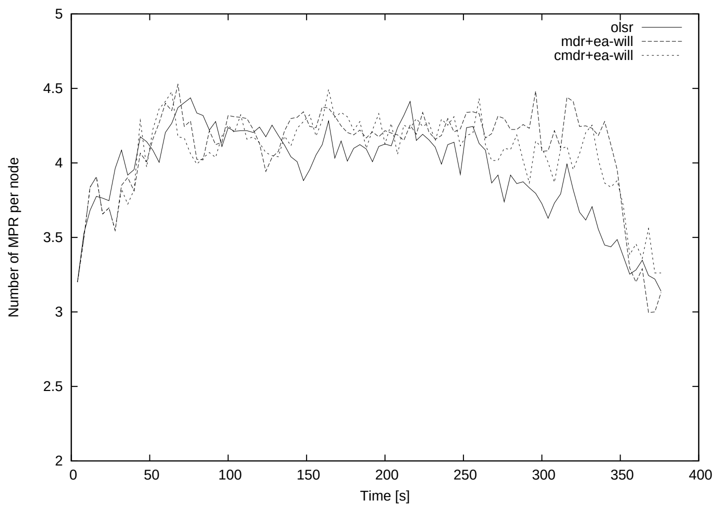

Fig. 7.37. Number of MPR per node over time, at 10 m/s

The last figure illustrates the aggregate throughput values over time, for different protocols: in terms of data delivery, energy-aware metrics outperform classical OLSR even in the case of a high mobility rate.

Performance vs Speed

In this subsection, the performances of the different simulated protocols with different mobility rates are compared.

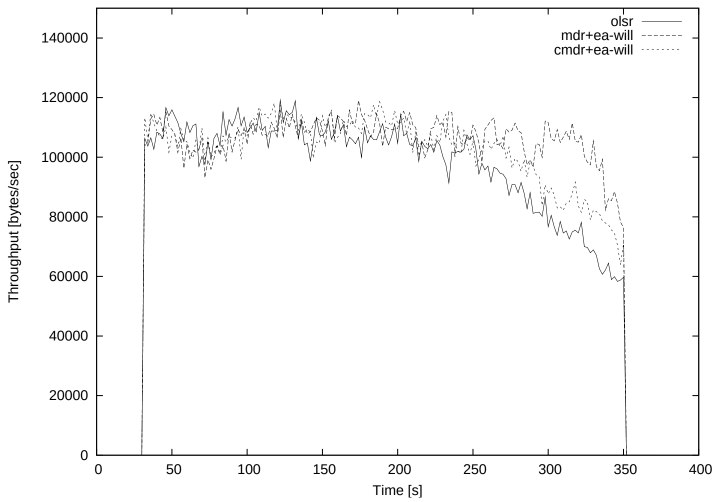

Fig. 7.38. Aggregate throughput over time, at 10 m/s

Figure 7.39 compares the values of minimum connection duration for all simulated protocols, with different nodes' maximum speeds. A higher value means a longer duration of the network before partitioning. From that point of view, MDR appears to be the best choice.

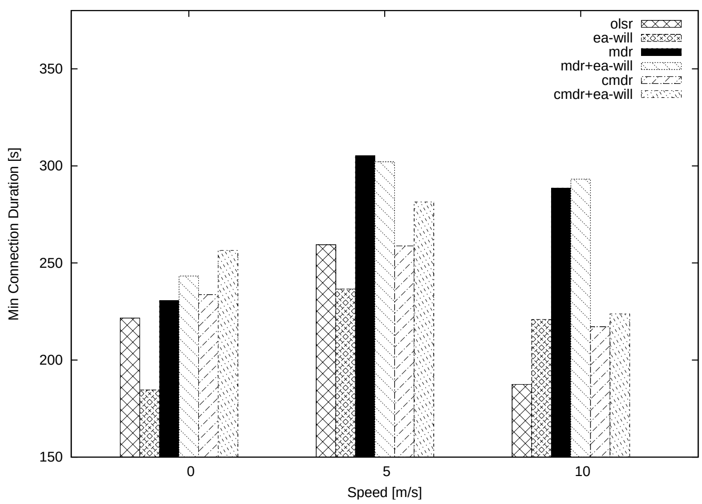

Fig. 7.39. Minimum connection duration, at different speeds

Figure 7.40 illustrates the values of minimum node lifetime for the different cases that have been simulated. This represents the time before the first node in the network ends its battery energy. Again, it is possible to highlight the good performance of MDR metric, especially when used in conjunction with EA-Willingness mechanism. Using CMDR, on the other hand, gives little advantages compared with classical OLSR protocol.

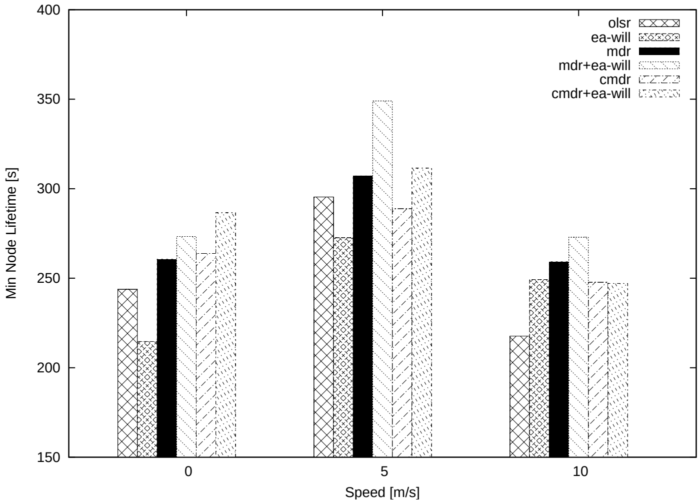

Fig. 7.40. Minimum node lifetime, at different speeds

The value of average final energy of nodes, depicted in Figure 7.41, measures the overall energy consumption of the various protocols: from that point of view, classical OLSR and CMDR metric are very similar. MDR metric, instead, shows higher energy consumption compared with other protocols, and the difference between them increases with node mobility.

Finally, Figure 7.42 illustrates the total number of data packet delivered to the destination, for all the simulations. It can be clearly noticed how energy-aware metrics, especially with the EA-Willingness setting, obtain better results even with increasing mobility. MDR and CMDR outperform classical OLSR protocol from this point of view.

Table 7.14 shows average data of end-to-end delay for all the cases. The values for energy-aware metrics are of the same order of magnitude as for OLSR protocol.

Table 7.14. End-to-end delay (ms)

Speed (m/s)	olsr	ea-will	mdr	mdr+ea-will	cmdr	cmdr+ea-will
0	3.47	3.50	4.09	4.12	3.61	3.90
5	10.35	8.66	8.42	10.26	6.22	8.23
10	13.03	11.63	11.40	9.96	14.71	13.76

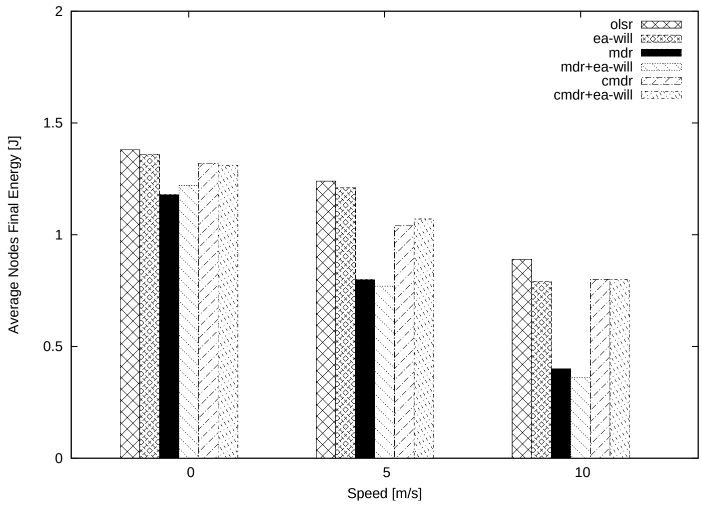

Fig. 7.41. Average nodes final energy, at different speeds

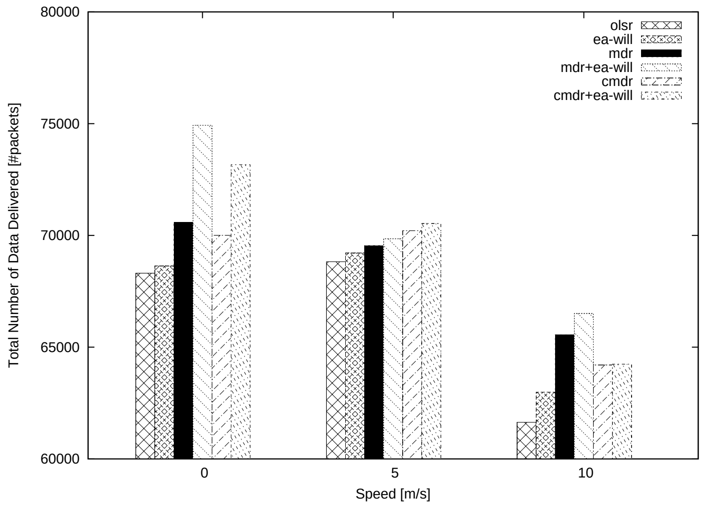

Fig. 7.42. Total number of data packets delivered, at different speeds

The data of the protocol overhead, also, gives values very similar to classical OLSR protocol, shown in Table 7.15. These two tables demonstrate that using the energy-aware metrics and mechanisms proposed does not have any cost in terms of network performance.

Table 7.15. Normalized Control Protocol Overhead (% bytes)

Speed (m/s)	olsr	ea-will	mdr	mdr+ea-will	cmdr	cmdr+ea-will
0	7.69	8.22	7.58	8.24	7.71	8.22
5	11.09	11.31	10.50	11.06	10.54	11.04
10	13.40	13.72	12.57	13.37	13.32	13.81

7.2.3 MDR Tuning

Constant consumption

In the case of constant energy consumption, the situation depicted in Figure 7.43 occurs. That picture shows clearly how the calculated drain rate always tends to its real value, and the time of convergence depends on the α value: with low values, convergence is faster. The difference is great: while $\alpha = 0.1$ reaches the DR value in the second interval (after 2 seconds), with $\alpha = 0.9$ it takes very longer (up to 100 seconds, or 50 intervals, to be near to the desired value).

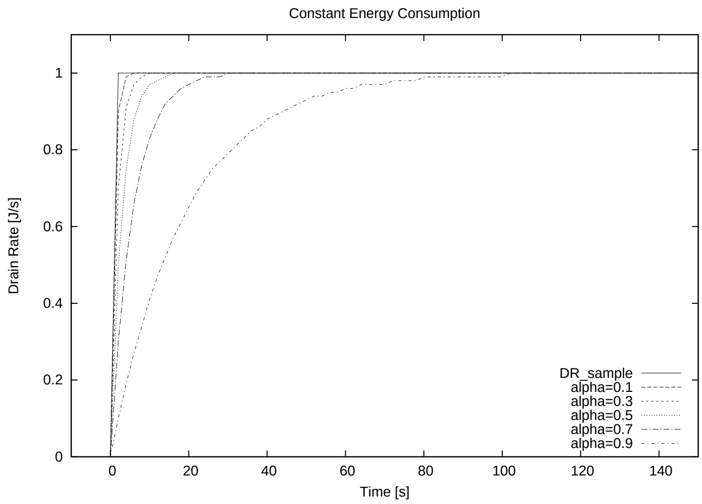

Fig. 7.43. Drain Rate estimations, with constant energy consumption and different values of α parameter

In the case of constant consumption values, then, very low values of α might be set: this leads to almost immediate convergence to the actual value of drain rate at the node.

On/off consumption, without residuals

In the case of variable energy consumption, if the on/off period and evaluation window are compatible there will be a situation very similar to the previous one. To show this case, it was plotted with a larger zoom, in Figure 7.44. The data of drain rate calculation are the same as with a constant consumption, since the node sees exactly the same drain rate in every evaluation interval. As can be noticed in the figure, while the energy consumption is variable, the drain rate is always the same, and the node behaves in the same way.

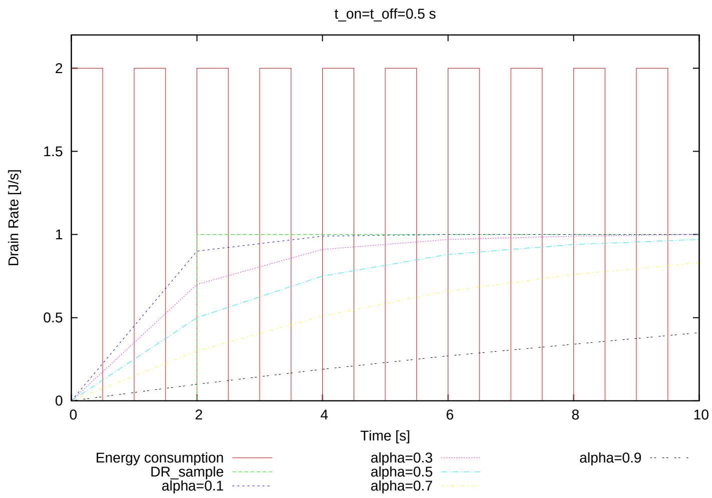

Fig. 7.44. Drain Rate estimations, with on/off energy consumption ($T = 1s$) and different values of α parameter

Then, low α values will again be preferred, to obtain a faster convergence of the drain rate estimation.

On/off consumption, with residuals

In real networks, the situation depicted in Figure 7.45 is most likely. Here, the drain rate samples measured by the node at each interval are variable, even if the overall consumption rate with time is fixed. Notice that the drain rate, even if oscillating around a fixed value, can be considered constant with time: in this case, a drain rate value near to the average value might be preferred, to different oscillating values changing at every evaluation interval.

A case has just been found in which a high value of best fits to energy consumption pattern: its slow convergence, in facts, allows the node to evaluate its own average energy consumption, rather than its actual value (variable and oscillating at every interval).

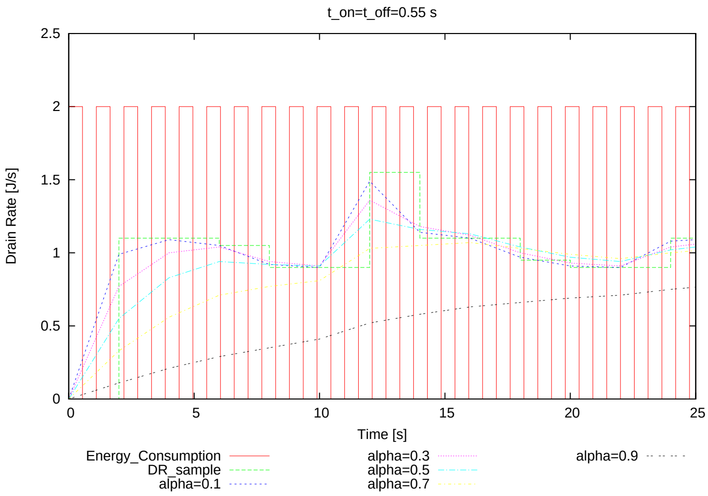

Fig. 7.45. Drain Rate estimations, with on/off energy consumption ($T = 1.1s$) and different values of α parameter

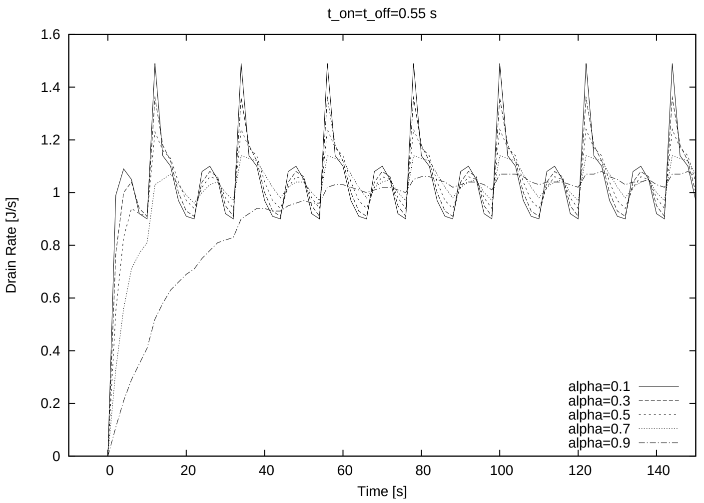

Fig. 7.46. Drain Rate estimations, with on/off energy consumption ($T = 1.1s$) and different values of α parameter, in a longer time scale

Figure 7.46, plotting the same situation in a longer time scale, shows well the long-term difference between different values of the parameter: high values are smoother and can better represent a consumption variable over the time, but oscillating around an average value.

On/off consumption, with interval

In another scenario, an on/off consumption pattern (with a period T of 0.6 seconds) and an interruption of 20 seconds in it was simulated. In this case, an α parameter is needed able to converge rapidly (not to declare any consumption within the interval) and to detect the average consumption value in the estimation windows. Figure 7.47 shows the Drain Rate estimation made by different, static α values.

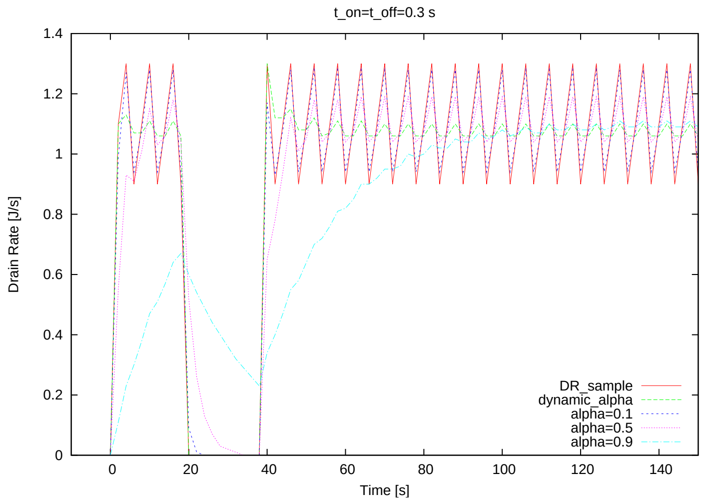

Fig. 7.47. Drain Rate estimations, with on/off energy consumption ($T = 0.6s$), an interval of no consumption and different values of α parameter

As one can notice, each value has its drawback: while low values cannot detect the average value of the on/off period, high values are not able to report correctly the inactivity interval.

Applying the adaptive model proposed in Section 6.3 to the traffic pattern seen in the last case, the estimations depicted in Figure 7.48 were obtained.

The adaptive α value, as can be seen, allows the declaration of Drain Rate values rapidly converging either to real values (in the case of constant energy consumption) and to average values (in the case of oscillating samples). The use of this parameter can significantly improve the performance of a mechanism like the Minimum Drain Rate, raising its effectiveness in a routing protocol for mobile ad-hoc networks.

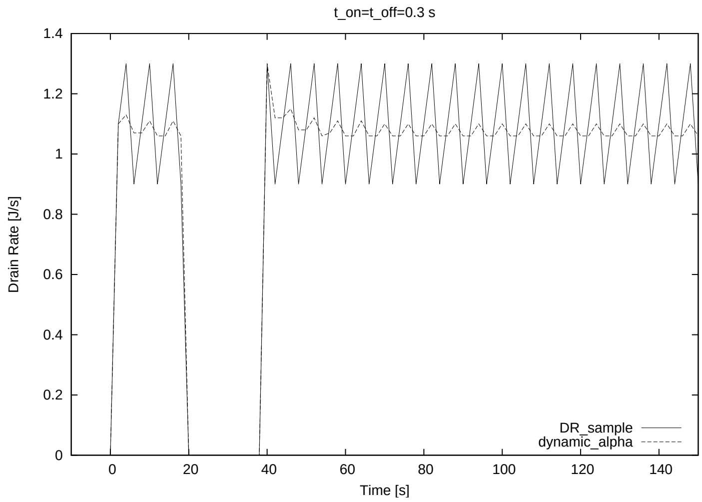

Fig. 7.48. Drain Rate estimations, with on/off energy consumption ($T = 0.6s$), an interval of no consumption and adaptive α parameter

Realistic case

To illustrate the effects of our adaptive model in a more realistic scenario, a node within a MANET was also simulated: its energy consumption with time is not regular, owing to transmission, reception and overhearing of packets. In Figure 7.49 this scenario is shown, with the approximation made by our adaptive α parameter selection.

In this scenario, too, the ability can be noticed of this approach to rapidly converge (in the case of significant changes in energy consumption) and to identify the average value of consumption (in the presence of rapidly varying samples).

7.2.4 Clustered Geographical OLSR

Comparison with the proactive approach

Figure 7.50 illustrates the advantages of the hybrid protocol in a scenario of 60 mobile nodes. The hybrid solution, in fact, consumes less energy than classical OLSR protocol, and the difference between the two approaches in terms of energy consumption increases with the simulation time.

Figure 7.51 traces the lifetime of the nodes during the simulation. The number of nodes in the network is 60.

It can be noticed how the use of the hybrid protocol leads to a longer lifetime for the network. The first nodes finish their batteries at 260 s, while with OLSR protocol some nodes go out of energy at 240 s.

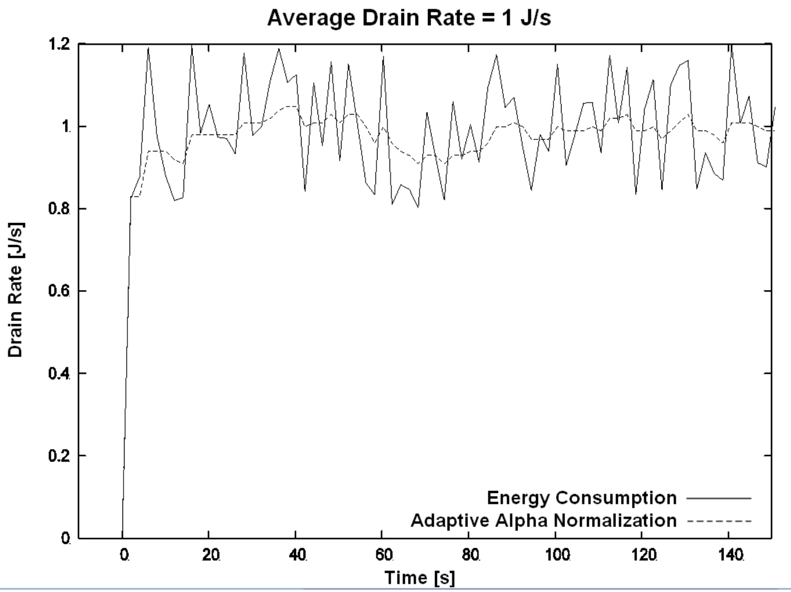

Fig. 7.49. Drain Rate estimations, with realistic energy consumption, using the adaptive α parameter

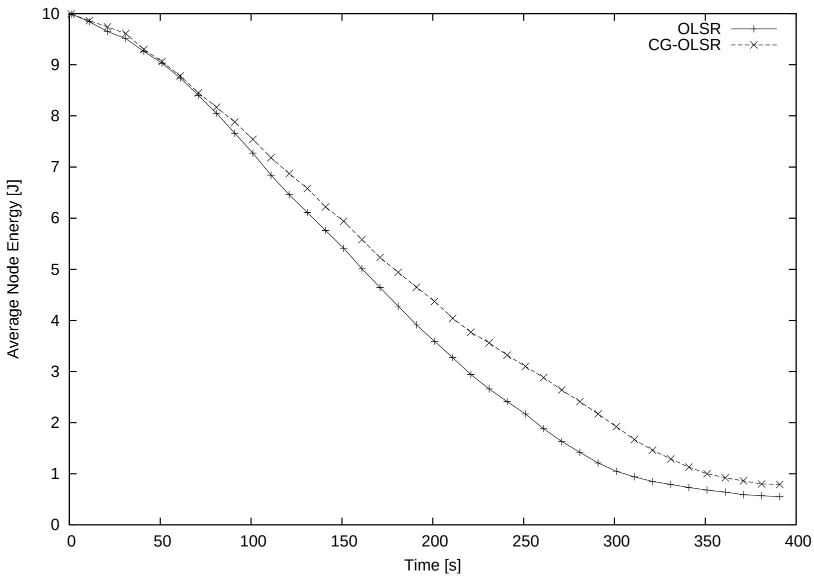

Fig. 7.50. Average nodes energy vs Time

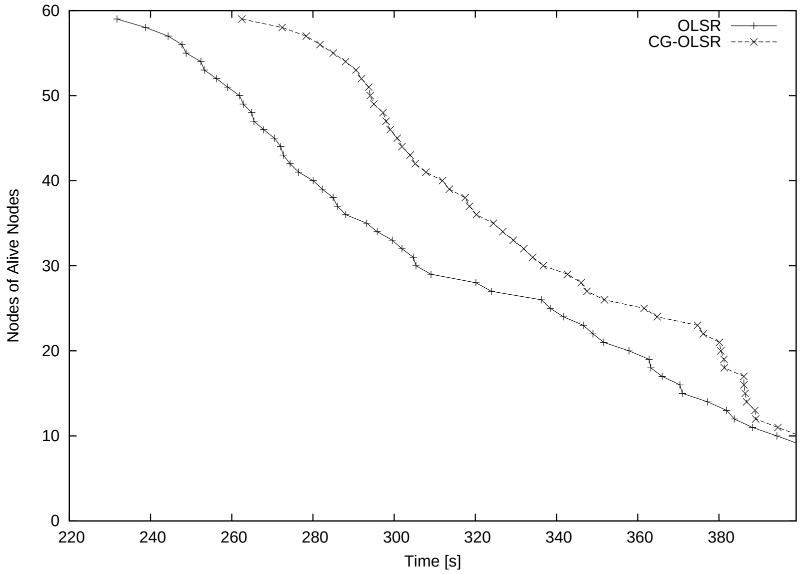

Fig. 7.51. Nodes lifetime

The Packet Delivery Ratio (PDR) is illustrated in percentage in Figure 7.52, comparing the different numbers of nodes simulated.

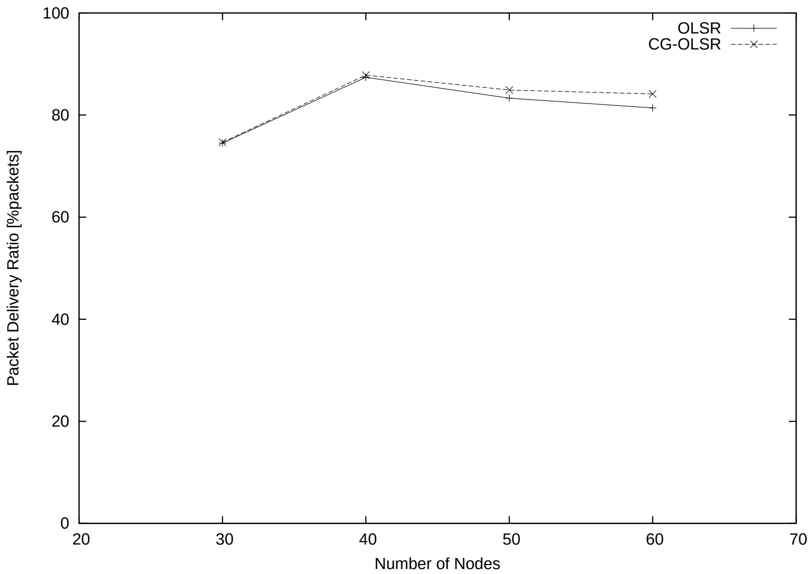

Fig. 7.52. Packet delivery ratio vs Number of nodes

The Normalized Control Protocol Overhead, expressed in bytes, is illustrated in Figure 7.53, with different numbers of nodes.

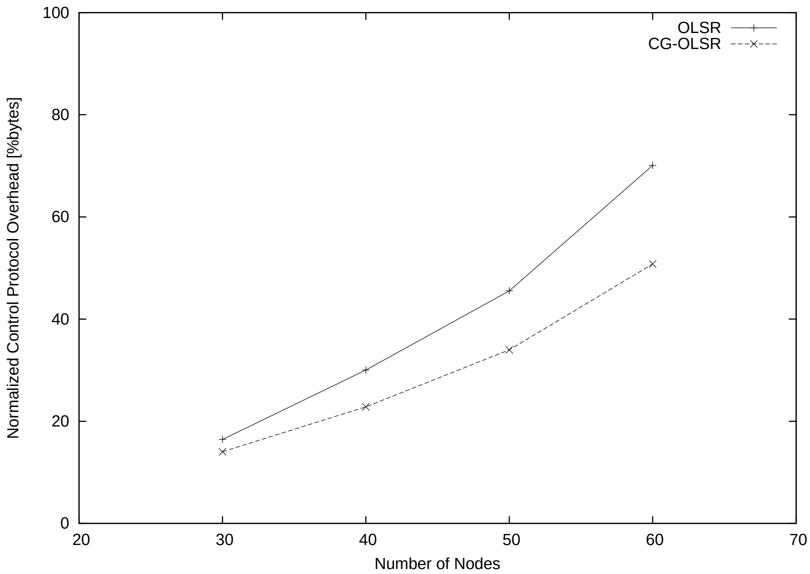

Fig. 7.53. Overhead vs Number of nodes

The use of the hybrid protocol reduces the overhead, thanks to the geographical information gathered by the nodes instead of the full network topology interchanged in OLSR.

In Figure 7.54 the throughput over the simulation is depicted, expressed in $bytes/sec$.

Traffic volume influence

To test the influence of the volume of traffic in the network over the performance of the hybrid protocol proposed, the simulations were repeated varying the number of traffic sources in the network.

Figure 7.55 shows the Packet Delivery Ratio with 6, 10 and 14 data traffic sources.

In Figure 7.56 the Normalized control protocol overhead is reported. As expected, this value decrements with the increment of the traffic in the network (as it defines a ratio between the control and the data information exchanged in the network).

Finally, Figure 7.57 illustrates the lifetime of the nodes over the time, with different traffic source numbers. Since the additional traffic in the network consumes more energy, we notice a longer lifetime when there are less traffic sources in the network.

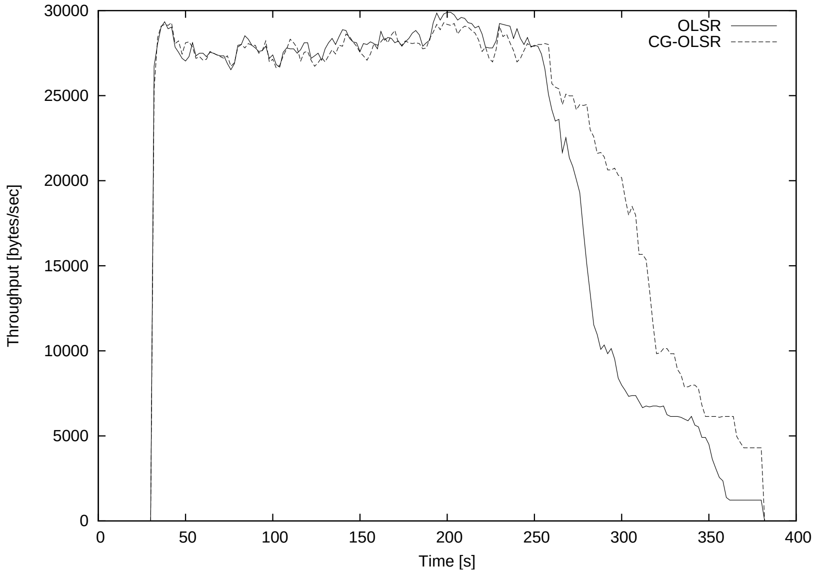

Fig. 7.54. Throughput vs Time

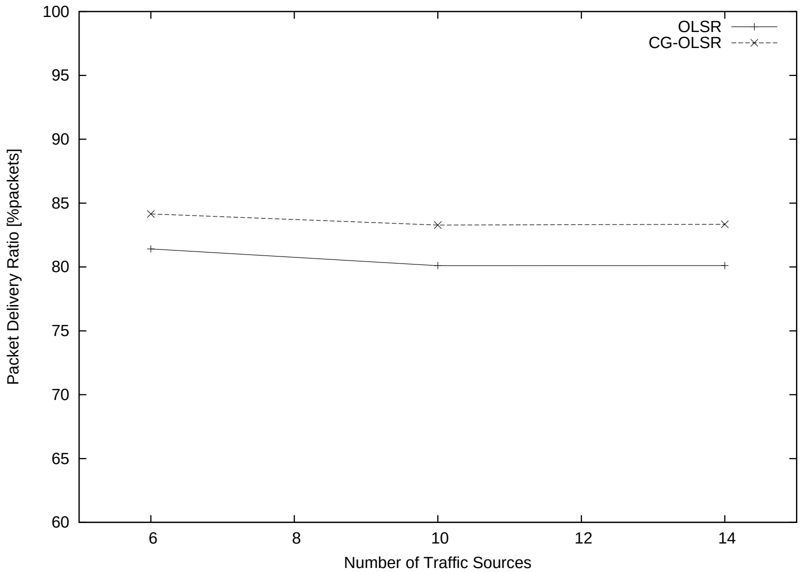

Fig. 7.55. Packet delivery ratio vs Traffic sources

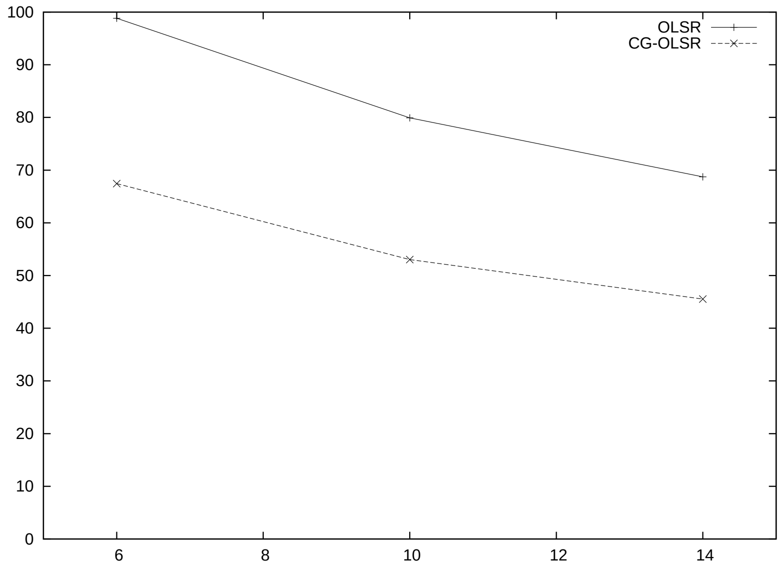

Fig. 7.56. Normalized Control Overhead vs Traffic sources

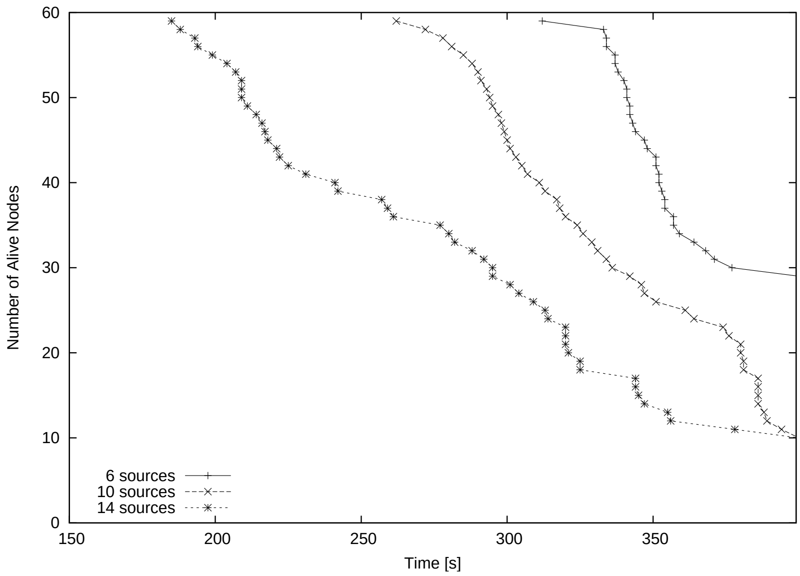

Fig. 7.57. Nodes lifetime with 8, 10 and 12 traffic sources

8

Summary and Conclusions

In this research thesis, the optimization of routing protocols for MANETs (a very important research area in the last few years) has been analysed from an energetic point of view. The work was particularly focused on OLSR protocol, a well known proactive protocol for ad-hoc networks, widely investigated by the telecommunication research groups and object of a RFC of the IETF MANET group.

The first part of the thesis focused on actual technology for the routing in a MANET, summarizing the main approaches proposed in literature for the multi-hop distributed routing, describing in details the OLSR protocol specification and highlighting the energy issues related to the mobile ad-hoc networks.

In the second part of the work, my personal contribution to that research area was illustrated. The research methodologies and instruments adopted were summarized, before the explanation of the actual proposals made for the improvement of the energy behavior of the OLSR protocol: a series of energy-aware metrics and mechanisms have been presented, an adaptive tuning for different energy consumption patterns has been illustrated, and a geographical clusterization aimed to reduce the overhead (and thus the energy consumption) has been proposed.

Finally, in the third part of the thesis, the actual results of the adoption of the energy-aware proposals in a wide range of simulation scenarios has been illustrated by means of graphics and tables. The simulations were intended to validate the actual value of the approaches proposed in various use cases, to highlight the strenght and the weakness points of each proposal, to put in evidence the benefits of using energy-aware approaches in a modern mobile network.

The obtained results are positive and promising: even if the proactive approach tends to consume more energy than a reactive one (esecially in low-trafficated networks), the adoption of the energy-aware proposals can improve significantly its behavior and make this kind on routing adapt to mobile networks. It can bring to MANETs the advantages of a link state protocol (reduced delay, quality of service), overcoming the limits of this kind of networks (limited bandwidth and energy constraints). Moreover, it has been proven that the introduction of energy-aware mechanisms in OLSR can improve its behavior without loss of performance (in terms of throughput and packet delivery).

In conclusion, this work represents a deep analysis of the energy issues related to MANETs in general, a complete summary of the state of the art in terms of energy optimization of ad-hoc routing protocols, and a concrete step towards the design of an efficient, proactive and adaptive routing protocol based on the OLSR. If this research field will continue to have the attention of the research community, this can be the effective basis for the spreading of the future mobile networks, interconnecting the increasing number of mobile devices able to communicate.

8.1 Future Work

The wide range of simulations conducted in this work put in vidence the advantages of the energy-aware mechanisms applied to a proactive routing protocol for ad-hoc networks. The experiments conducted, moreover, suggest that there is the possibility to obtain even better results by the study and the efficient definition of protocols tailored to the features of modern mobile networks. The implementation of adaptive mechanisms can lead to manage the complex dynamics of such networks, and the adoption of an efficient clusterization scheme can make the solutions more scalable and performant.

The solutions proposed in this work need to be validated in a wider range of scenarios (in terms of mobility, density and traffic), and to be effectively implemented in mobile devices. Moreover, the specification of OLSR can be extended with the mechanisms proposed, and other similar mechanisms can be proposed in order to make of OLSR the main protocol for the mobile personal communications of the future to come.

Appendices

A

Tcl Script for Simulations

```
#---------------------------------------------------------------
# Sample file for OLSR simulation
#---------------------------------------------------------------

#
#Load external functions
#
source cbr-source.tcl ;#CBR/UDP sources of traffic

source 802_11g.tcl ;#802.11g (54 Mbps)

#---------------------------------------------------------------
# Create a simulation, with wireless support.
#---------------------------------------------------------------
set ns_ [new Simulator]

set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
set val(ifq) Queue/DropTail/PriQueue
set val(ifqlen) 50  ;# max packet in ifq
set val(ll) LL
set val(ant) Antenna/OmniAntenna

set val(mac) Mac/802_11
set val(rp) OLSR
set val(seed) 0.0

set val(x) 500  ;# X dimension of the topography
set val(y) 500  ;# Y dimension of the topography
```

```
set val(n) 15  ;# how many nodes are simulated
set val(v) 2  ;# speed of mobile nodes

set val(em) "EnergyModel"
set val(etx) 1.4 ;# Watts
set val(erx) 1.2 ;# Watts
set val(eid) 0.9 ;# Watts
set val(ein) 50 ;# Joules
set val(eov) true ;# overhearing energy consumption:
   # true or false

set val(t)  50  ;# simulation time
set val(st) "FCP"  ;# cbr traffic type:
   # FCP (fixed connection pattern) or
   # VCP (variable connection pattern)
set val(sn) 4  ;# number of traffic sources
set val(si) 10  ;# start of traffic
set val(se) 45  ;# end of traffic
set val(sp) 512  ;# data packet dimension, in bytes
set val(sr) 20  ;# send rate (packets per second)

set val(rt)  4  ;# routing type: 0(MTPR), 1(MBCR),
   # 2(MMBCR), 3(CMMBCR), 4(MDR)
set val(rl)  false  ;# use of link layer notifications:
   #true or false
set val(rw)  false  ;# energy-aware willingness setting:
   #true or false

switch $val(rt) {
 0 { set type "MTPR" }
 1 { set type "MBCR" }
 2 { set type "MMBCR" }
 3 { set type "CMMBCR" }
 4 { set type "MDR" }
}

Phy/WirelessPhy set overhear_ $val(eov)

Agent/OLSR set routing_type_ $val(rt)
Agent/OLSR set use_mac_ $val(rl)
Agent/OLSR set will_setting_ $val(rw)

set val(outputfile) "tr-pOLSR.tr"
$ns_ use-newtrace ;
set tracefd [open $val(outputfile) w]
```

```
$ns_ trace-all $tracefd

set val(outputnamfile) "nam-pOLSR.nam"
set namtrace [open $val(outputnamfile) w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
set god_ [create-god $val(n)]

#
# create channel
#

set chan_1 [new $val(chan)]

$ns_ node-config -adhocRouting $val(rp) \
 -llType $val(ll) \
 -macType $val(mac) \
 -ifqType $val(ifq) \
 -ifqLen $val(ifqlen) \
 -antType $val(ant) \
 -propType $val(prop) \
 -phyType $val(netif) \
 -channel $chan_1 \
 -topoInstance $topo \
 -energyModel $val(em) \
 -initialEnergy $val(ein) \
 -txPower $val(etx) \
 -rxPower $val(erx) \
 -idlePower $val(eid) \
 -agentTrace ON \
 -routerTrace ON \
 -macTrace OFF \
 -movementTrace OFF

#-----------------------------------------------------------
# Create nodes with OLSR agent
#-----------------------------------------------------------

set rng [new RNG] ;#random number generator
$rng seed 0.0

for {set i 0} {$i < $val(n)} {incr i} {
 set node_($i) [$ns_ node]
```

```
 $node_($i) random-motion 0 ;# disable random motion
}

#
# Define traffic model
#
puts "Loading connection pattern..."

set interval [expr 1.0/$val(sr)]  ;# send interval (s)

if { $val(st) == "FCP" } {
 for {set i 0} {$i < 2*$val(sn) } {incr i; incr i} {
  cbr-source [expr ($i)%($val(n))]
   [expr ($i+1)%($val(n))] $val(si) $val(se)
   2 $i $val(sp) $interval
 }
}
if { $val(st) == "VCP" } {
 # changes on connection pattern, every 10 seconds
 set changes [expr ($val(se)-$val(si))/10]
 set start $val(si)
 set stop [expr $start+10]
 for {set i 0} {$i < $changes } {incr i} {
  for {set j 0} {$j < $val(sn)} {incr j} {
   # selects randomly a source node
   set n1 [expr floor([$rng uniform 0 $val(n)])]
   # selects randomly a destination node
   set n2 [expr floor([$rng uniform 0 $val(n)])]
   cbr-source [expr $n1%($val(n))]
    [expr $n2%($val(n))] $start $stop
    2 $j $val(sp) $interval
  }
  set start $stop
  set stop [expr $start+10]
 }
}

#
# Define node movement model
#
puts "Loading scenario file..."

exec ./setdest_mig -n $val(n) -p 0 -s $val(v) -t $val(t)
 -x $val(x) -y $val(y) >
 sc-n$val(n)-s$val(v)-t$val(t)-x$val(x)-y$val(y).sc
```

```
source "sc-n$val(n)-s$val(v)-t$val(t)-x$val(x)-y$val(y).sc"

#-----------------------------------------------------------
# Finishing procedure
#-----------------------------------------------------------

proc finishSimulation { } {
    global ns_ node_ val

 for {set i 0} {$i < $val(n)} {incr i} {
  $node_($i) log-energy-2
 }

    # Exit
    puts "Finished simulation."
    $ns_ halt
}

#-----------------------------------------------------------
# Run the simulation
#-----------------------------------------------------------

proc runSimulation {duration} {
    global ns_ finishSimulation
    for {set j 3.0}{$j < $duration}{set j [expr $j + 3 ]}{
 $ns_ at $j "puts t=$j"
    }
    $ns_ at $duration "finishSimulation"
    $ns_ run
}

runSimulation $val(t)

#-----------------------------------------------------------
```

B

Perl Script for Tracefile Analysis

```
#
# usage: perl-Statistics.pl [TRACEFILE]
# produces output file: txt-Statistics-TRACEFILE.txt
#

if($ARGV[0] ne "") {$trace=$ARGV[0];}
else {
 print "Usage: perl-Statistics.pl <TRACEFILE>\n";
 exit(-1);
}

open(trfile,"$trace") || die "Couldn't open trace file\n";

$trace =~ s/tr-//;
$trace =~ s/\.tr//;

open(out,">txt-Statistics-$trace.txt") ||
  die "Couldn't open output file\n";
print out "\#FILE txt-Statistics-$trace\n";

$agtsent=0; # num.of packets sent at agent level
$agtrcv=0; # num.of packets received at router level

$control=0; # num.of control packets sent
$controlbytes=0; #num.of control bytes sent

%sendtime={}; #send time of packets
%rcvtime={}; #receive time of packets

$e2edelay=0; # e2edelay
```

```
$maxdelay=0; # max e2e delay value

$numforwards=0;  # n. of fwd (hops) of received packets
$optforwards=0;  # optimal n. of fwd for a packet

# drop reasons (and number of occurrences)

$drops{END}=0;  # END_OF_SIMULATION
$drops{COL}=0;  # MAC_COLLISION
$drops{DUP}=0;  # MAC_DUPLICATE
$drops{ERR}=0;  # MAC_PACKET_ERROR
$drops{RET}=0;  # MAC_RETRY_COUNT_EXCEEDED
$drops{STA}=0;  # MAC_INVALID_STATE
$drops{BSY}=0;  # MAC_BUSY
$drops{DST}=0;  # MAC_INVALID_DST
$drops{SLP}=0;  # MAC_SLEEP (smac sleep state)

$drops{NRTE}=0; # RTR_NO_ROUTE (no route)
$drops{LOOP}=0; # RTR_ROUTE_LOOP (routing loop)
$drops{TTL}=0;  # RTR_TTL (ttl reached zero)
$drops{TOUT}=0; # RTR_QTIMEOUT (packet expired)
$drops{CBK}=0;  # RTR_MAC_CALLBACK (MAC callback)
$drops{SAL}=0;  # RTR_SALVAGE

$drops{IFQ}=0;  # IFQ_QFULL (no buffer space in IFQ)
$drops{ARP}=0;  # IFQ_ARP_FULL (dropped by ARP)
$drops{FIL}=0;  # IFQ_FILTER
$drops{OUT}=0;  # OUTSIDE_SUBNET (dropped by base stations)

while (<trfile>) {
 if (/^d/){
  @vals = split(" ",$_);
  $drops{$vals[20]}+=1;  #-Nw (drop reason)
 }
 elsif (/AGT/){
  if (/^s/) {
   $agtsent++;
   @vals = split(" ",$_);
   $sendtime{$vals[40]}=$vals[2]; #sndtime(packetID)=time
  } elsif (/^r/) {
   $agtrcv++;
   @vals = split(" ",$_);
   $rcvtime{$vals[40]}=$vals[2]; #rcvtime(packetID)=time
   $numforwards+=$vals[48];  #-Pf (n. of forwards)
   if($vals[50]<10000000)
```

```
    $optforwards+=$vals[50]; #-Po (optimal n.of fwd)
   }
  } elsif (/RTR/ && /^s/ && (/-It DSR/ | /-It OLSR/)) {
   $control++;
   @vals = split(" ",$_);
   $controlbytes+=$vals[36]; #-Il (packet length)
  }
}

$rcvbytes=$agtrcv*512;

@pkts=keys %rcvtime; #all received packets
$n=0;
$del=0;
foreach (@pkts){
 $del=$rcvtime{$_}-$sendtime{$_};
 if ($del > $maxdelay){
  $maxdelay=$del;
 }
 $e2edelay+=$del; #finally, total e2e delay
 $n++; #num.of packets for average e2e delay calculation
}
$e2edelay=$e2edelay/$n; #average e2e delay

print out "AVERAGE VALUES\n";
if($agtsent==0){
 $pktratio=0.0;
}else{
 $pktratio=$agtrcv/$agtsent*100.0;
}
print out "Data Packet Delivery Ratio [%]\t$pktratio\n";
$e2emsec=$e2edelay*1000.0;
$maxdelay=$maxdelay*1000.0;
print out "E2E Delay [ms]\t$e2emsec\n";
print out "Max E2E Delay [ms]\t$maxdelay\n";
if($agtrcv==0){
 $pktoverhead=0.0;
}else{
 $pktoverhead=$control/$agtrcv*100.0;
}
print out "Control Protocol Overhead [packets]\t
   $control\n";
print out "Normalized Control Protocol Overhead [% packets]\t
   $pktoverhead\n";
if($rcvbytes==0){
```

```
 $byteoverhead=0.0;
}else{
 $byteoverhead=$controlbytes/$rcvbytes*100.0;
}
print out "Control Protocol Overhead [bytes]\t
   $controlbytes\n";
print out "Normalized Control Protocol Overhead [% bytes]\t
   $byteoverhead\n";
if($agtrcv==0){
 $numforwards=0.0;
 $optforwards=0.0;
}else{
 $numforwards=$numforwards/$agtrcv;
 $optforwards=$optforwards/$agtrcv;
}
print out "Average Number of Forwards [hops]\t
   $numforwards\n";
print out "Average Optimal Path Length [hops]\t
   $optforwards\n";
print out "Total number of data sent [packets]\t
   $agtsent\n";
@reasons=sort keys %drops; #all drop reasons
print out "_Drop Reason_\t_Number_\n";
foreach (@reasons){
 print out "$_\t$drops{$_}\n";
}

close(trfile);
close(out);
```

References

1. Internet Engineering Task Force, Manet working group charter. Available from: http://www.ietf.org/html.charters/manet-charter.html
2. Fotino M, Gozzi A, Cano JC, Calafate C, De Rango F, Marano S, Manzoni P (2007) Evaluating Energy-aware Behavior of Proactive and Reactive Routing Protocols for Mobile Ad Hoc Networks. Proc. 10th International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS07), 16-18 July, San Diego, CA, USA
3. IEEE 802.11 Standard. LAN MAN Standards Committee of the IEEE Computer Society, 1999
4. Royer E, Toh CK (1999) A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications Magazine 6 (2) (1999)
5. Hong X, Xu K, Gerla M, Los Angeles CA (2002) Scalable routing protocols for mobile ad hoc networks Network, IEEE, vol. 16, no. 4, pp. 11-21, 2002.
6. Perkins CE et al. (2003) Ad Hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, July 2003. http://www.ietf.org/rfc/rfc3561.txt
7. Johnson DB et al. (2002) The dynamic source routing protocol for mobile ad hoc networks DSR. Internet Draft, February 2002
8. Toh CK (1997) Associativity based routing for ad hoc mobile networks. Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems 4 (2) (1997)
9. Perkins CE, Bhagwat P (1994) Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, Computer Communications Review, pp.234-244, October 1994
10. Murthy S, Garcia-Luna-Aceves JJ (1996) An efcient routing protocol for wireless networks. ACM/Baltzer Journal on Mobile Networks and Applications, Special Issue on Routing in Mobile Communication Networks, vol. 1, pp. 183-197, October 1996
11. Pei G, Gerla M, Chen TW (2000) Fisheye state routing: a routing scheme for ad hoc wireless networks. Proc. IEEE International Conference on Communications (ICC), vol. 1, pp. 7074, June 2000
12. Clausen T, Jacquet P (2003) Optimized Link State Routing Protocol (OLSR). RFC 3626, October 2003. http://www.ietf.org/rfc/rfc3626.txt

13. Pearlman M, Haas ZJ (1999) Determining the optimal conguration for the zone routing protocol. IEEE Journal on Selected Areas in Communications, vol. 17, pp. 13951414, August 1999
14. Joa-Ng M, Lu IT (1999) A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. IEEE Journal on Selected Areas in Communications, vol. 17, pp. 14151425, August 1999
15. Karp B, Kung HT (2000) Greedy Perimeter Stateless Routing for Wireless Networks. Proc. 6th Annual ACM/IEEE International Conference Mobile Computing Networks, pp.243-254, Boston, August 2000
16. Feeney L, Nilsson M (2001) Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment. Proc. IEEE INFOCOM, Anchorage, AK, 2001
17. Allard G, Minet P, Nguyen DQ, Shresta N (2006) Evaluation of the energy consumption in MANET. Adhoc-Now 2006, Ottawa, Canada, August 2006
18. Toh CK (2001) Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks. IEEE Communication Magazine
19. Kunz T (2008) Energy-Efficient Manet Routing: Ideal vs Realistic Performance. 2008
20. Kim D, Garcia-Luna-Aceves JJ, Obraczka K, Cano JC, Manzoni P (2003) Routing Mechanisms for Mobile Ad Hoc Networks Based on the Energy Drain Rate. IEEE Transactions on Mobile Computing, Vol. 2, No. 2, January 2003, pp. 161-173
21. Misra A, Banerjee S (2002) MRPC: Maximizing network lifetime for reliable routing in wireless environments. Proc. IEEE WCNC, 2002, pp. 800-806
22. Chiasserini CF, Rao RR (2000) Routing protocols to maximize battery efficiency. Proc. MILCOM, IEEE, 2000, pp. 496-500
23. Cardei M, Wu J, Yang S (2004) Topology Control in Ad hoc Wireless Networks with Hitch-hiking. Proc. IEEE SECON 2004, October 2004
24. Ingelrest F, Simplot-Ryl D, Stojmenovic I (2006) Optimal Transmission Radius for Energy Efficient Broadcasting Protocols in Ad Hoc Networks. Proc. IEEE Transactions on Parallel and Distributed Systems, June 2006
25. Xu Y, Heidemann J, Estrin D (2001) Geography-informed Energy Conservation for Ad Hoc Routing. Proc. ACM Mobile Computer and Networking Conference, July 2001, pp. 70-84
26. Jin X, Cai W, Zhang Y (2005) A RED based minimum energy routing algorithm for wireless ad-hoc networks. Proc. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, September 2005, pp. 757-761
27. Jones CE, Sivalingam KM, Agrawal P, Chen JC (2001) A survey of energy efcient network protocols for wireless networks. Wireless Network Journal, 17 (4) (2001) 343-358
28. Lindsey S, Sivalingam K, Raghavendra CS (2001) Power optimization in routing protocols for wireless and mobile networks. Handbook of Wireless Networks and Mobile Computing, Wiley, 2001
29. Taddia C, Giovanardi A, Mazzini G, Zorzi M (2005) Energy efcient unicast routing protocols over 802.11b. Proc. IEEE Global Telecommunications Conference (GLOBECOM05), 28 November-2 December 2005
30. Wan PJ, Calinescu G, Li XY, Frieder O (2001) Minimum-energy broadcast routing in static ad hoc wireless networks. Proc. IEEE INFOCOM, Anchorage, AK, 2001

31. Jung S, Hundewale N, Zelikovsky A (2005) Energy efficiency of load balancing in MANET routing protocols. Proc. 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and 1st ACIS International Workshop on Self-Assembling Wireless Networks, May 2005, pp. 476-483
32. Senouci S, Naimi M (2005) New routing for balanced energy consumption in mobile ad hoc networks. Proc. 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, October 2005, pp. 238-241
33. Garcia JE, Kallel A, Kyamakya K, Jobman K, Cano JC, Manzoni P (2003) A novel DSR-based energy-efcient routing algorithm for mobile ad-hoc networks. Proc. 58th IEEE Vehicular Technology Conference (VTC Fall. 2003), Orlando, Florida, USA, 6-9 October 2003, pp. 2849-2854
34. Luo Y, Wang J, Chen S (2006) An energy-efficient DSR routing protocol based on mobility prediction. Proc. International Symposium on a World of Wireless, Mobile and Multimedia Networks, June 2006
35. Kwon S, Shroff NB (2006) Energy-Efficient Interference-Based Routing for Multi-hop Wireless Networks. Proc. IEEE INFOCOM 2006, Barcelona, Spain, April 2006
36. Benslimane A, El Khoury R, El Azouzi R, Pierre S (2006) Energy Power-Aware Routing in OLSR Protocol. Proc. 1st International Mobile Computer and Wireless Communication Conference, September 2006, pp. 14-19
37. Ghanem N, Boumerdassi S, Renault E (2005) New energy saving mechanisms for mobile ad-hoc networks using OLSR. Proc. 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, October 2005, pp. 273-274
38. Guo Z, Malakooti B (2007) Energy Aware Proactive MANET Routing with Prediction on Energy Consumption. Proc. 2007 International Conference on Wireless Algorithms, Systems and Applications, August 2007, pp. 287-293
39. Heinzelman W, Chandrakasan A, Balakrishnan H (2000) Energy-efficient communication protocol for wireless microsensor networks. HICSS 2000, Maui, Hawaii, USA, vol. 2, January 2000
40. Xia D, Vlajic N (2007) Near-optimal node clustering in wireless sensor networks for environment monitoring. Proc. AINA 2007, Niagara Falls, Ontario, Canada, May 2007
41. Ganesan D, Govindan R, Shenker S, Estrin D (2001) Highly-resilient, energy-efficient multipath routing in wireless sensor networks. ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), volume 1, no 2, 2001
42. Shah RC, Rabaey JM (2002) Energy Aware Routing for Low Energy Ad Hoc Sensor Networks. IEEE WCNC 2002, Volume 1, pp. 17-21, March 2002
43. Srinivas A, Modiano E (2003) Minimum Energy Disjoint Path Routing in Wireless Ad-Hoc Networks. Proc. MOBICOM03, September 2003
44. Hassanein H, Luo J (2006) Reliable Energy Aware Routing In Wireless Sensor networks. Proc. IEEE DSSNS 2006, April 2006
45. Shresta N (2006) Reception Awareness for Energy Conservation in Ad Hoc Networks. PhD Thesis, Macquarie University Sydney, Australia, November 2006
46. De RangoF, Fotino M, Marano S (2008) EE-OLSR: Energy Efficient OLSR Routing Protocol for Mobile Ad-hoc Networks. Proc. Milcom08

47. Woo K, Yu C, Lee D, Youn H, Lee B (2001) Non-blocking, localized routing algorithm for balanced energy consumption in mobile ad hoc networks. Proc. MSCOTS, ACM/IEEE, 2001, pp.117-124
48. Vassileva N, Barcelo-Arroyo F (2008) A Survey of Routing Protocols for Maximizing the Lifetime of Ad Hoc Wireless Networks. International Journal of Software Engineering and Its Applications, Vol. 2, No. 3, July, 2008
49. McCabe A, Fredin M, Cullen A, Axelsson L (2005) A power consumption study of DSR and OLSR. Proc. IEEE Military Communications Conference (MILCOM05), 17-20 October 2005, pp.1954-1960
50. Zhao Q, Tong L (2005) Energy efciency of large-scale wireless networks: proactive versus reactive networking. IEEE Journal on Selected Areas in Communications, 23 (5) (2005) 11001113
51. Ryu JH, Song S, Cho DH (2001) New Clustering Schemes for Energy Conservation in Two-Tiered Mobile Ad-Hoc Networks. Proc. IEEE ICC01, vo1. 3, June 2001, pp. 862-66
52. Cao L, Dahlberg T, Wang Y (2007) Performance evaluation of energy efficient ad hoc routing protocols. Proc. IEEE IPCCC, 2007, pp. 306-313
53. ns-2 Network Simulator. http://www.isi.edu/nsnam/ns
54. VINT Project: A Collaboration among USC/ISI, Xerox Parc, LBNL and UCB. http://www.isi.edu/nsnam/vint/index.html

List of Publications

Journals:

De Rango F., Cano J.-C., Fotino M., Calafate C., Manzoni P., Marano S.: "OLSR vs DSR: A comparative analysis of proactive and reactive mechanisms from an energetic point of view in wireless ad hoc networks", Computer Communications Journal (Elsevier) 31(16): 3843-3854, 2008

Conference Proceedings:

Fotino M., Gozzi A., De Rango F., Marano S., Cano J.-C., Calafate C., Manzoni P., "Evaluating Energy-aware behaviour of proactive and reactive routing protocols for mobile ad hoc networks", in Proceedings of the 10th International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'07), San Diego, CA, USA, 16/18 July 2007

Fotino M., Gozzi A., Cano J.-C., Calafate C., De Rango F., Manzoni P., Marano S., "Evaluating Energy Consumption of Proactive and Reactive Routing Protocols in a MANET", in Proceedings of the 1st International Conference on Wireless Sensor and Actor Networks (WSAN‘07), Albacete, Spain, 24/26 September 2007

De Rango F., Fotino M., Marano S., "EE-OLSR: Energy Efficient OLSR Routing Protocol for Mobile Ad-hoc Networks", in Proceedings of Military Communications Conference (MILCOM’08), San Diego, CA, USA, 17/19 November 2008

De Rango F., Fotino M., "Energy Efficient OLSR Performance Evaluation under Energy Aware Metrics", in Proceedings of the 12th international conference on Symposium on Performance Evaluation of Computer & Telecommunication Systems (SPECTS’09), p.193-198, Istanbul, Turkey, 13/16 July 2009

De Rango F., Fotino M., "Adaptive Tuning in Minimum Drain Rate Algorithm for Routing Protocols in MANETs", in Proceedings of the 21th Personal, Indoor and Mobile Radio Conference (PIMRC’10), Istanbul, Turkey, 26/29 September 2010