Dynamic Remote Access Solution to a Hot-Zone

by

Bachar Wehbi

Engineering diploma, Lebanese University, 2004

Thesis

Submitted in partial fulfillment of the requirements
for the degree of Research Master in Computer Science
at the University of Pierre et Marie Curie, 2005

Supervised by……..……

Virginie Galtier
Associate Professor

September 2005

Dynamic Remote Access Solution to a Hot-Zone

by

Bachar Wehbi

Abstract

Although wireless hot spots and Mesh networks are now widely deployed, their transmission range could be one of their essential limitations. On the other hand, recent studies have identified the scalability limitations of pure ad hoc networks. Shared access to a single wireless channel in the presence of a dense network and long paths leads to congestion and high channel contention, thus degraded performance. In this project we are investigating the possibility to combine stable fixed Mesh networks with multi hop ad hoc extensions. The resulted hybrid network allows extending the hot-zone of the Mesh network by permitting farther nodes to reach the Mesh by relaying on intermediate nodes, at the same time avoids the direct ad hoc long communications by using the Mesh routers. We present our "EcoMesh" architecture and its requirements in term of dynamic address configuration, routing and mobility support. Address assignment is performed by extending the DHCP protocol to multi-hop environment. We propose a routing protocol with an integrated mobility support specifically tailored for the architecture. We show through some tests that allowing multi-hops provides a high gain in term of covered zone.

Project Supervisor: Virginie Galtier
Title: Associate professor

Solution d'Accès Distant Dynamique à une « Hot-Zone »

par
Bachar Wehbi

Résumé

Les points d'accès WIFI ainsi que les solutions de réseaux maillés sont de plus en plus déployés, pourtant leur portée de transmission assez limitée peut poser des problèmes. En même temps, des études récentes ont montrés les problèmes de passage à l'échelle des réseaux ad hoc purs. Le partage d'un seul canal radio par un large nombre d'utilisateurs et les communications distantes augmentent la congestion et par suite dégradent les performances. Dans ce projet, nous étudions la possibilité de combiner les réseaux fixes maillés avec des réseaux ad hoc réduits à quelques sauts. Le réseau hybride résultant permet d'étendre la zone de couverture de la « hot-zone » du réseau maillé en permettant aux utilisateurs éloignés d'utiliser les nœuds intermédiaires comme relais réseau. En même temps, cela permet d'éviter d'avoir des longues communications de manière purement ad hoc par l'utilisation de l'infrastructure maillée disponible. Nous présentons dans ce projet cette architecture nommée « EcoMesh » et ses exigences en termes d'attribution d'adresses, routage et support de mobilité. La configuration d'adresses est atteinte par un simple mécanisme qui étend DHCP pour les environnements multi-sauts. Nous proposons un protocole de routage spécifiquement conçu pour cette architecture. Un mécanisme supportant la mobilité est intégré avec le routage. Nous démontrons à travers des tests que cette extension multi-sauts fournit un gain élevé en terme de zone de couverture.

Sous la direction de: Virginie Galtier
Titre: Maître de Conférences

Acknowledgments

I am indebted to my supervisor, Virginie Galtier, for her high standards, her advice and guidance in preparing this project. I would like also to thank my research group, the EcoMesh team, for their discussions and suggestions.

Thanks to Gerardo, Wissam and Mounir for the endless supply of entertainment and support; I looked forward to coming to lab every day.

Finally, thanks to Baba, Mama, Sara, Ahmad and Rola for their love and encouragement. This work would not have been possible without the aforementioned individuals.

Table of contents

List of figures

List of tables

Chapter 1

INTRODUCTION

In the last few years the deployment of IEEE 802.11 WLANs has grown exponentially in areas such as airports, university campuses, hotels and enterprises. These infrastructure based networks allow mobile users to access network services and the Internet without the need for wires. The high data rates achievable for 802.11a, b and g far surpass that of wide-area cellular networks, however, the essential limitation of 802.11 WLAN networks is their low transmission range reaching only 250m in open environments compared with up to 20km for cellular networks. In addition, in 802.11 infrastructure mode, each access point has to be connected to the wired infrastructure; access points act just like wireless switches.

A Mesh network is constituted of fixed wireless routers equipped by two or more interfaces; the wireless user-side interface that allows for mobile nodes to connect to the Mesh network, and a backbone wired or wireless interface that allows the Mesh routers to be interconnected forming the hot-zone. The advantage of Mesh networks is their learning capacities about the topology and their ability to dynamically construct and maintain routes to other mesh routers. The Mesh networks allow for easy and flexible deployment and maintenance. Many Mesh routers could share one connection to the internet or any external network as opposed to WLAN access points. The research community was interested in the Mesh capabilities and many papers were recently published and the IEEE 802.11s working group is working on the standardization of Mesh within 802.11.

As opposed to infrastructure based or fixed wireless Mesh networks, Mobile ad-hoc networks consist of a collection of wireless equipped mobile nodes that auto organize their environment in a distributed fashion without the need for an established stable infrastructure. Each ad-hoc node plays a double role and acts as a terminal and as a router for forwarding packets on behalf of other nodes, supporting multi-hop communications. Pure ad hoc nodes have high overhead and scalability problems, performances degrade asymptotically with higher communication ranges making it impossible to deploy large scale ad hoc networks.

In this project we are investigating the possibility to combine stable fixed Mesh networks with ad hoc extensions. The resulted hybrid network allows extending the hot-zone of the Mesh network by permitting farther nodes to reach the Mesh by relaying on intermediate nodes. The deployment of such hybrid network requires solving some points. Dynamic address configuration must be considered to assign users with IP addresses. Routing should take into account the hybrid architecture of the network so ad hoc communications could be permitted only for limited range, thus avoiding ad hoc

scalability problems, and the mesh infrastructure could be exploited for distant communications and for accessing to the internet. Finally, mobility support should be taken into consideration in such a dynamic environment so users could move within the network without loosing connectivity or experiencing long disconnection periods.

The rest of this report is organized as follows. Chapter 2 presents the EcoMesh project and the requirements and architecture of its network. Chapter 3 addresses the related work and a comparative study on address configuration and presents our adapted solution for our hybrid network's context. In chapter 4, and after presenting our routing and mobility requirements and a background study, we describe our routing and mobility protocol totally adapted to the EcoMesh context. Our simulation environment and some test results are presented in chapter 5. Finally, in chapter 6, we draw some conclusions and future work.

Chapter 2

THE ECOMESH[1] PROJECT

This chapter presents the EcoMesh project that proposes to combine Mesh and ad hoc networks in a collaborative environment. It aims to show that hybrid networks would be useful in many situations and that by adapting the networking components, this network could be deployed.

Section 2.1 presents the project and the goals behind it and some situations where it could provide high benefits. Then it describes the specific scenario we are placed in where an ISP is planning to extend its Mesh network. Section 2.2 identifies the requirements of the so called "EcoMesh like network" in term of address configuration routing protocol and mobility support. Finally, section 2.3 presents the EcoMesh network architecture and the entities composing it.

2.1 EcoMesh overview

The key idea of the EcoMesh project is to exploit the multi hop capabilities of ad-hoc networks to extend the coverage zone of a Mesh network's "hot-zone". The obtained network would have multi benefits; first it represents an economical solution to increase the limited coverage of 802.11 based Mesh routers (even possibly WLAN access points), second it provides the possibility to deploy a complexity reduced ad-hoc network by limiting its diameter to a few hops. In this hybrid network, closer users have to share their resources (bandwidth, battery, computation power) with the community and serve as an intermediate between the mesh network and farther users. This collaborative behavior represents the vital factor in the existence of such a hybrid network; therefore the EcoMesh project proposes to associate a reward policy with the routing protocol to incite users for collaboration.

This kind of hybrid networks would be of great interest in different possible scenarios. We could imagine it as:

- Extension of an enterprise network reserved for employees. An enterprise that has deployed a Mesh network to cover its site may leave, for economical reasons, uncovered zones that are less dense (for example the airport surface). In this case a user will meet probably the same users; the intra ad-hoc communications will be considerable and we have not to care about the collaboration as it could be imposed by the enterprise policy. The concern is the possibility to offer an acceptable quality of service for such a professional

[1] "Projet à credit incitatif" financed by the GET. http://ecomesh.objectis.net

environment.

- Extension of a campus network reserved for the personnel and students use. In this case too, the probability to meet the same users is high, thus the intra ad-hoc communications will be important. An incentive mechanism has to be adopted; however the prior knowledge between users would enforce the collaboration motivation.
- Extension of an association's network for non lucrative use. For example the city citizens may put for open access their internet connections. In this case, the obtained network is not administered, and their will be probably no prior knowledge between ad hoc arriving users. The usage of the network will be different for "permanent" users than for "ad hoc" users. Permanent users may use this network to inter-communicate (we could imagine a voice service that could replace the telephone) as they share a prior knowledge and to access to the internet; in contrast, ad hoc users will probably use the network only to access to the internet. The main concern here is to how to motivate users to collaborate in such un-administered and open network especially when we take into consideration the possibility for a user to change its identity (MAC, IP, Nickname …) for malicious use.
- Extension of ISPs network reserved for clients use. In this case, the probability to meet the same users will be low, thus intra ad-hoc communications would probably be minor, and the main network usage will be to access to the internet or simply outside the ad-hoc extension. Each user could be identified by a sort of a client ID, so the ISP will have the ability to track who is cooperating and adopt the appropriate incentive policy; on the other hand, as the service is paid, a minimum service quality have to be guaranteed. A question to answer would be: what is the minimum density and collaboration that offer a given service threshold?

In the EcoMesh project we fixed our studies for the case of an internet ISP who is intending to extend its Meshed hot-zone's coverage area by a mean of spontaneous collaborative ad-hoc extension reserved for its own clients use. Distant clients will be able to reach the meshed side by sharing bandwidth resources offered by intermediate clients acting as relays. In this scenario, we can assume that the probability to meet the same users is very low, therefore the intra ad-hoc communications are minor compared to the communications through the Meshed borne (Internet access, extranet or any services offered by the ISP…). Accordingly, the network will have the Mesh backbone as a stable part that will be always reachable; if not the ad-hoc extension will loose its reason to be. This kind of hybrid networks is not well studied in the literature.

Our work is to find appropriate solutions and/or to adapt already existing ones for the problems created by the characteristics of this kind of hybrid networks in order to make it possible to extend in an efficient manner the Mesh zone by an ad hoc extension. Our work is limited to the ad hoc extension. We suppose that the Mesh side is connected in an efficient and consistent manner that is independent from the ad hoc extension. How routing is performed on the Mesh level is not part of our study, our concern is to give the Mesh backbone the required information to support the presence of ad hoc nodes more than one hop away (as we will see in chapter 4 this complete separation is impossible so we were obliged to make some requirements on the Mesh level).

2.2 Requirements of an EcoMesh like network

The dynamic remote access solution to an EcoMesh like hybrid network has to answer the following requirements:

- Automatic address configuration: network nodes have to be assigned with valid IP addresses in order to participate in the network operation. IP address assignment has to be automatic and assigned addresses have to be unique within the network and globally identified in order to allow for internet access. The address assignment time should be realistic, nodes have not to wait a long period before being assigned and the mechanism should have limited communication overhead.

- Adapted routing protocol: as mentioned above, the ad-hoc extension will be limited to few hops and the communications will be minor between ad-hoc nodes, rather it will be concentrated between ad-hoc nodes and the internet or simply through the mesh bone. Thus the routing protocol should be adapted to these characteristics.

- Mobility support: the location of a node is needed in order to deliver packets to and from the node. In wireless environments, users are mobile and may change of location at any time. The location information should be updated whenever a node changes its point of attachment. In traditional one hop wireless networks the location track was possible because of the direct association (at the physical and MAC layers) between the mobile node and the access point or mesh router. In multi hop environment, MAC layer information are nor sufficient to track the mobility of users. A new approach should be studied that permit to track the mobility of multi hops away mobile users.

2.3 The EcoMesh architecture

The EcoMesh like network is composed of two entities: Ad-hoc nodes and Mesh routers. An ad-hoc node is equipped with a wireless interface that allows it to communicate with other ad-hoc nodes and the Mesh routers. A Mesh router is equipped with two interfaces, one wireless interface for communicating with ad-hoc nodes and the other could be a wireless or wired interface for communicating with other Mesh routers. If the second interface is wireless, it is assumed that it operates on a different frequency range and possibly on different technology. The Meshed backbone is totally connected.
The ad-hoc range in the EcoMesh context is limited to a maximum of "*K*" hops. An ad-hoc node can't directly communicate with another ad-hoc node neither with the Mesh routers if they are more than *K* hops away. To reach an ad-hoc node that is more than *K* hops away, the communication must pass by a Mesh router if it exists at a distance less than *K* hops. The figure 2.1 represents a representative EcoMesh network connected to the internet and constituted of 2 Mesh routers and 8 ad-hoc nodes. The connection between the routers is not represented in the figure. If we suppose that 2 is the maximum allowed ad-hoc hops, then the nodes A, B, C, D, E, F and G are able to reach the Mesh backbone and participate in the network functioning. Note that node H is not considered as part of the network as it is 3 hops away than the nearest Mesh router. Node C can

communicate with node F by taking per example the route C – A – R1 – E – F that comports 4 "ad-hoc" hops; note that the route C – A – D – E – F is not allowed as it comports more than 2 direct ad hoc hops. A guarantee of the restriction of the ad-hoc range to a maximum of *K* is that every ad-hoc node can be reached by no more than 2*K* wireless hops (connection between C and F through R1 per example).

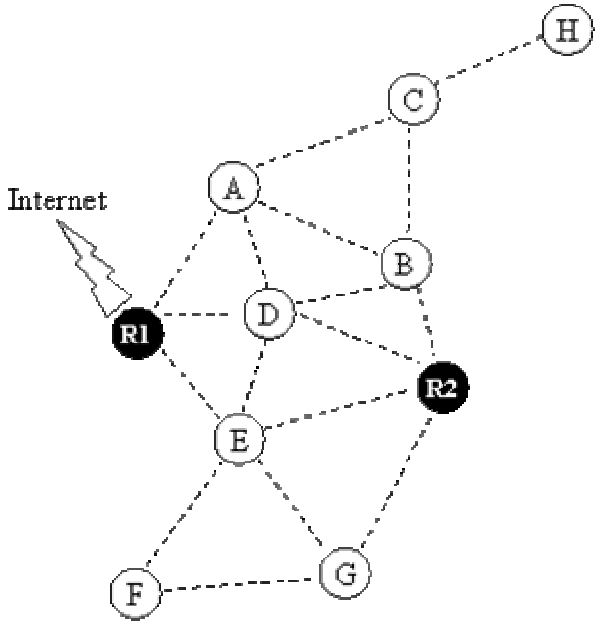


Figure 2.1: A representative EcoMesh network

Ad-hoc nodes take advantage of the nearby Mesh routers to maintain connectivity, and use the nearest one as default gateway. Thus, ad-hoc nodes have to learn by some mechanism the mesh routers less than K hops away. In figure 2.1, G is at 1 hop from R2 and at 2 hops from R1; to communicate with C per example, G must pass by R2 and not by R1. In other words, a node has to use always the nearest Mesh router to reach farther node or to access to the internet.

Mesh routers have to maintain a list of ad-hoc nodes that use it as a default gateway to avoid the need for flooding the network searching for a specific user and to let it possible to implement a roaming mechanism as we will see in the chapter 4.

2.4 Summary

This chapter presented the "EcoMesh like" hybrid network composed of a Mesh backbone to which is attached spontaneous ad hoc extensions. This hybrid network architecture could be seen as the compromise between the limited coverage problem of 802.11 based Mesh routers and the scalability problem of pure ad hoc networks. An ISP could exploit this architecture to extend its coverage area with limited investment. Some issues like address configuration, routing and mobility support have to be addressed in order to make it possible to deploy this architecture.

Chapter 3

ADDRESS CONFIGURATION IN ECOMESH

Address configuration is a required phase before network nodes could communicate. For traditional wired and wireless networks, this issue was dealt with by the introduction of Dynamic Host Configuration Protocol (DHCP) for administered address configuration and the dynamic configuration of IPv4 Link-Local Addresses (IPv4LLA) for auto assignment of IP addresses.

For Ad Hoc networks, neither approach is suited; DHCP is too centralized for such a dynamic environment and IPv4LLA assumes a local broadcast network. That's why new address autoconfiguration approaches must be adopted for Ad Hoc networks.

This chapter lists the state of the art of address configuration in ad-hoc networks, and compares these approaches according to the context and requirements of the EcoMesh context. It describes then our adopted address configuration solution that is most suited for our environment.

Section 3.1 and 3.2 begin with an overview of address configuration approaches in traditional wired networks and the constraints that make it impossible to directly extend these approaches to multi-hop wireless networks. Section 3.3 dresses the state of the art of address configuration approaches in wireless multi-hop networks classified into stateful, stateless and hybrid approaches according to their management of the address space. Section 3.4 presents first the requirements of address configuration in the EcoMesh context, and then draws a mature and complete comparative study of existing approaches. This comparison is based on a long list of metrics of high importance in our scenario (overhead, latency, sensitivity on losses, address uniqueness and stability…) and could be used to plan for address configuration in multi-hop wireless networks. According to this study, section 3.5 describes in detail the reasons behind adopting the "All DHCP Relay" approach for the EcoMesh. "All DHCP Relay" is a promising address configuration mechanism for hybrid wireless networks like EcoMesh, it's based on the DHCP protocol (it needs no modifications) so it's implementation ready.

3.1 Traditional Address Configuration Approaches

3.1.1 Dynamic Host Configuration Protocol (DHCP)

DHCP [1] is the first mechanism proposed for dynamically assigning IP addresses. It is based on a client/server architecture where a central entity, the DHCP server, is responsible for assigning IPs for requesting nodes and maintaining the state for each address of the available address range, thus address duplication is totally avoided.

When a new node starts and has no IP address configured, it broadcasts a message to discover if a DHCP server is present. If a DHCP exists, it replies to inform the new node (DHCP client) of its presence. Then the DHCP client requests directly the DHCP server for an IP address, the DHCP server picks a free IP of its pool and sends it to the client who confirms its reception of the offer.

The message exchange between the DHCP server and the DHCP client are identified by the MAC addresses thus a DHCP server should exist locally. To overcome this limitation, a DHCP relay could be used in local networks where no DHCP server is present. The DHCP relay acts as an intermediate between the server and the client; it intercepts the client broadcasted messages and sends them in unicast to the server to allow messages to cross routers; thus it should be configured with the IP address of the server.

3.1.2 Dynamic Configuration of Link Local Addresses (Zeroconf.)

A DHCP infrastructure is not suitable in case of dynamic networks where centralizing the address configuration is not appropriate. That's why the Zeroconf. working group has proposed a mechanism [2] to allow nodes to auto-configure with link local addresses in the range of 169.254/16. This approach applies to environments where the network is built to allow only local communications with no global connection to the internet or an external network.

This protocol is suitable for communication between nodes within the same MAC broadcast domain. When a node joins the network, it randomly chooses an IP address and sends an ARP (Address Resolution Protocol) message destined to the chosen address. If the IP address is already used, the new node will receive a message indicating so, and then it chooses another address and restarts the procedure. If the new node receives nothing, it concludes that the IP is free so it can use it.

3.2 Constraints in Multi-hop Wireless Networks

In contrast to wired LAN networks, where a broadcast message is able to reach all nodes on the link, the wireless multi-hop networks are characterized by a multi hop topology. Thus even a broadcast message should be routed from hop to hop. That's why traditional autoconfiguration protocols like DHCP and Zeroconf could not be directly applicable.

Another issue in wireless multi-hop networks is the energy and bandwidth constraints. Wireless nodes have in general limited power supply and need to keep control communication overhead at minimum. The broadcast nature of the wireless medium and the interference between simultaneous communications make the packet loss relatively high leading to higher packet retransmission and as a result higher power and bandwidth consumption and higher communication delays.

3.3 Classification of Address Assignment Approaches

To overcome the limitations of traditional address configuration protocols in wireless

multi-hop networks, many approaches have been proposed in the last few years (focusing on ad hoc networks). These address assignment approaches could be classified as stateful or stateless approaches according to the management of the address space. For stateful approaches, the state of each address is held in such a way the network have a vision of assigned and non assigned IPs, so address duplication could be avoided. For stateless approaches, each node randomly chooses its own address and performs a duplicate address detection test to ensure that the chosen address is not already used.

3.3.1 Stateful approaches

All stateful approaches maintain address allocation tables to track assigned and free addresses, so existing nodes can easily assign unused addresses to requesting nodes. The challenge for stateful approaches is to synchronize the allocation tables to ensure that any used address figures in the allocation table. The advantage of stateful approaches is the duplicate free assignment.

In Agent Based Addressing [3] only one node, the Address Agent (AA), holds a centralized allocation table and is allowed to assign addresses to requesting nodes, thus it should be always reachable. The allocation table contains already assigned IP addresses with their corresponding MAC addresses and lifetimes. It is designed for IPv6 and supposes the uniqueness of MAC addresses.

The AA periodically floods the network with the allocation table in a form of “Verify” messages to indicate its presence and to allow for already assigned nodes to confirm their leases. When an already configured node receives the “verify” message it should reply to refresh its lease lifetime. When a new node joins the network, it waits until it receives a verify message from the AA, then it requests by unicast (how an unassigned node could initiate a unicast communication is not specified) an IP address form the AA. The IP address is constructed based on the nodes and AAs MAC addresses. A mechanism for dynamically elect the AA is specified so that the network could survive in case of AA departure. In case of AA change, all assigned nodes have to request a new address from the new AA, this leads to unnecessary address changes.
To distinguish between different networks, the AA constructs a “Network ID” derived from its MAC address and floods it within the Verify packets.

In contrast with “Agent Based Addressing” where only one node is responsible for assigning addresses and maintaining the allocation table, MANETconf [4] is based on a “common distributed address table” where each node is able to assign IP addresses and maintains an allocation table that contains already allocated addresses and pending allocations. Thus, the synchronization of these distributed tables constitutes the most critical and complex task of this protocol.
When a new node joins the network, it searches for already assigned nodes by broadcasting (local broadcast does not need an IP address) a message to test its neighborhood. Then it chooses the first neighbor who replies as the initiator and contacts it to request an IP address. The initiator chooses a free IP from its allocation table and floods the whole network to have the permission to assign the chosen address. When a node receives this message, it consults its allocation table to see if the address is already assigned, and replies with a positive permission only if the address is free. This phase is

required for two reasons; first the different tables may not be totally synchronized because of the necessary synchronization convergence delay, second it is possible for two nodes to simultaneously choose the same IP to assign it to different arriving nodes.
If all existing nodes reply positively, the initiator concludes that the address is free and sends it to the requester and floods it in the network to confirm the address assignment and let all nodes update their tables. If one or many nodes reply negatively, the initiator concludes that the address is already assigned and repeats the procedure from the start. If the initiator detects that one or more nodes did not reply, it re-contacts them by unicast reclaiming their permission, and inform the network of the departure if the concerned node does not reply.
Differentiation between networks is based on a network ID which is a 2-tuple, the first is the lowest IP address in use in the network and the second a unique identifier generated by the node with the lowest IP address.

The idea behind Prophet [5] is that in place of maintaining an allocation table and working hard to synchronize it along the network, each node maintains a generation function and a state value to generate a sequence of numbers (addresses), thus address allocation is totally decentralized and generates zero traffic. The problem in this protocol corresponds then to choose the good generation function. Such a function should fulfill the following requirements:

- The interval between two occurrences of the same number in a sequence is extremely long.
- The probability that the function returns the same number for two different state values is very low.

These two conditions may be respected only if the address range is extremely high therefore it is not suited for IPv4. When a node joins the network, it searches for a configured neighbor by sending a local broadcast. If the new node receives many replies, it contacts one for requesting an IP address. The requested node uses its generation function and state value to obtain a new address and a state value and provides them to the requesting node. Then the initiator updates its state value to not generate the same number again. When a node leaves the network, address reclamation is not needed because the same number will reoccur in the sequence separated by a long period of time. For networks of moderate size, the authors propose a generation function "f(n)" based on a product of prime numbers with each prime raised to the power of the state value. If 'R' is the address space, then the generation function $f(n) = a + 2^{e1} * 3^{e2} * 5^{e3} * 7^{e4} \mod(R) + 1$. With 'a' the IP address of the node generating the new address. Unfortunately, there is no analytic proof that the described function fulfils the necessary requirements.

In the Buddy protocol [6], each node is responsible of a different allocation table constituted of a part of the whole address space and used to assign addresses for new comings. At the same time, each node holds the whole address table to keep track of the evolution of the network. Synchronization between all nodes is an essential part of the protocol to allow each node to build the whole address table.
At the beginning there is only one node that has the entire pool; this node detects no neighbors, thus it auto assigns with the first IP of the predefined address range. When a new node joins the network, it broadcasts a request message to its neighbors reclaiming

an IP address. If it receives one or more responses, it chooses the first who replies and sends it an address request, the requested initiator replies by dividing its own address pool and sends back the second half along with a copy of the address table. Then the new node assigns itself the first address in the pool and sends a confirm message to its initiator. The synchronization of the address table involves each node to periodically broadcast its address table. The detection of IP address leaks is accomplished by "buddy nodes", imagine A and B two buddy nodes (A: 0→31 and B: 32→63) to detect address leaks, A test B and vice versa. If one node detects that the other is missing, it merges the missing IP range with its own pool. To distinguish between different networks, a network is associated always with a network ID. The network ID is generated by the first node in the network.

The "All Relay Autoconfiguration" proposed in [7] is based on a simple idea that after been configured through DHCP, the client runs the DHCP relay agent module to allow farther nodes to reach the DHCP server. This approach is suited for ad-hoc networks that require connection to the internet and therefore are connected to a fixed borne. It supposes also that a centralized DHCP server is always reachable; it could be somewhere on the fixed borne.
As this proposal is addressed for ad-hoc networks that are connected to a fixed borne, the network will start by nodes that are one hop away. For these nodes the DHCP protocol works fine; the broadcast communications initiated by one hop away nodes are intercepted by the fixed borne and directed toward the DHCP server (for instance the server could be the wireless fixed borne or a dedicated server connected to the wireless fixed borne). The problem arises for nodes that are more than one hop away. To allow for these nodes to obtain an IP address, each node must have at least one DHCP relay agent present in its neighborhood. To guarantee this constraint it is required that all nodes in the ad hoc network run the DHCP relay module after obtaining their own address. The DHCP relay agent must be supplied with the IP address of the DHCP server; since the relay is already assigned with a valid address, it can extract the server's IP from the "server identifier" option field of the DHCP packets.

3.3.2 Stateless approaches

All stateless approaches are characterized by auto-allocation of IP addresses, which means, each node chooses randomly its IP address. Then the node should perform a mechanism for duplicate address detection to insure that its chosen IP is unique within the network. The challenge in stateless approaches is to detect in moderate delays and traffic, the potential address duplication. The advantage of stateless approaches is their relative simplicity compared to stateful approaches.

The Strong Duplicate Address Detection (SDAD) presented in [8] is the base for all stateless approaches. It consists of a simple mechanism that allows an ad hoc node to choose an IP address and test if it's already used or not. We can consider this proposal as an extension of the Zeroconf. for multi hope networks.
When a node initializes, it picks 2 addresses, a "temporary address" and a "tentative address" in the range 169.254/16 (0→2047 and 2048→65534 respectively). The temporary address is used only in the initialization phase as a source address for requests flooded to detect if the tentative address is already used or not.

The new node floods the network with an address request (ICMP) packet destined to the tentative address and waits a certain period of time. If during this period it receives a reply, it concludes that the address is already used and reinitiates the process. If during this period it receives nothing, it repeats the request with the same tentative address a specified number of times to insure that the address is not used before it releases the temporary address and definitively adopt the tentative address.

The main limitation of SDAD is that the duplicate address detection is only limited to the initialization phase. So, if for a reason of network losses or temporal disconnection the auto configuration process leads to address duplication, the network will not be able to solve this duplication which disturbs the performance of the network. The Weak Duplicate Address Detection (WDAD) protocol proposed in [9] aim to extend the duplicate address detection mechanism for the whole lifetime of the network. The idea is that duplicate addresses may be tolerated as long as packets reach the destination node intended by the sender, even if the destination node's address is also being used by another node. That's why each node selects an identification key at initialization phase and distributes it with its IP address in all routing messages to make routing capable of differentiating between potential duplicate IPs.
Each node maintains keys along with IP addresses in its routing table. When a node receives a routing message with an IP address that exists in its table, it checks if the keys are different. If they are different, a duplicate address is detected and the entry is marked as invalid and additional steps should be taken to inform other nodes about this duplication (steps not specified in the protocol).

Passive Duplicate Address Detection (PDAD) [10] is a duplicate detection mechanism designed for link state routing protocols. The idea behind PDAD is that instead of explicitly trying to detect and solve address duplication by sending control information, each node can investigate routing information and deduce address duplication from events that never occur in case of unique addresses but do occur if there are address duplicates.
With proactive routing, the nodes periodically flood the network to inform other nodes about their neighborhood. These control packets contain sequence numbers to distinguish between fresh and old packets. Based on this information, PDAD analyzes incoming routing packets to detect address duplicate. For example, sequence numbers are increased with each packet, and reset occurs once in a long period of time. Normally a node should not receive a message with its IP address as the source address and a sequence number greater than its own counter value. Accordingly, if it receives such a packet an address conflict had been detected. For a complete list of mechanisms used to passively detect address duplication refer to [10].

The Ad Hoc IP Address Autoconfiguration presented in [11] combines the mechanisms of SDAD and WDAD to accomplish address consistency. Thus the duplication detection mechanism not only checks for duplication during initialization, but also checks and resolves potential address duplication detected by intermediate nodes using routing messages. This fusion of the two mechanisms allows for smooth handling of network partition and merging. Each node must choose a 128 bits long key and appends it to control packets of routing protocol; intermediate nodes must maintain the key value for each address in routing table or cache. The autoconfiguration procedure is exactly the

same as described in SDAD. When a node receives a routing packet, it investigates all IP addresses and key values contained in that packet, and compares them to addresses and keys contained in its address table or cache. If for the same IP address it finds different key values, then an address conflict has occurred; the node in this case, must send in unicast an address error message indicating the occurrence of address conflict to the node with duplicate address associated with the smaller key value.
During normal operation, if a node receives an address error with duplicate address the same as its own address, the node releases its address and starts autoconfiguration from scratch in order to reconfigure with an unassigned IP address.

3.3.3 Hybrid approaches

Hybrid approaches tend to combine mechanisms from both stateless and stateful approaches, in order to improve reliability and scalability of address autoconfiguration. The price is more complex protocols.

The Hybrid Centralized Query-based Autoconfiguration (HCQA) protocol proposed in [12] is the first hybrid approach. It utilizes SDAD mechanism along with a centrally maintained allocation table in order to improve address consistency.
At initialization phase, a node chooses two addresses, a temporary and a tentative one, and performs SDAD exactly as explained in SDAD. If the address autoconfiguration was successful, the new node must register its tentative address with an "Address Authority". Therefore it waits for an advertisement of the AA a certain period of time. Upon receiving the advertisement, the new node launches a registration request and waits for the registration confirmation (ACK message). Only after the confirmation, the node may begin to use this address. After a successful registration, the node runs a timer and reinitiates the registration process each time the timer expires.
In addition to holding the states of all assigned IP addresses, the Address Authority can help in detecting address duplication in the initialization phase by replying to address request destined to a used tentative address. This is of high importance especially when the concerned node is temporary disconnected.
The protocol indicates a mechanism to dynamically elect the AA in case of AA departure or network partitioning. In addition the AA chooses a unique identifier (ex: its MAC address) and floods it along with the periodic advertisement messages in order to identify the network in case of network merge.

3.4 Autoconfiguration in the EcoMesh Context

In this part we will speak about address autoconfiguration in the context of the EcoMesh project. After defining the characteristics of the EcoMesh model we will compare and classify the existing approaches according to a group of characteristics derived from the EcoMesh context and according to some characteristics of wireless multi hop networks.

3.4.1 Requirements for EcoMesh's Address Autoconfiguration

When placed in the EcoMesh context, address autoconfiguration should fulfill the following requirements:

- Topology change: in the EcoMesh context, the clients may use the network for different purposes, they may use it to access the web, read their mails, and chat… thus their lifetime within the network may vary from client to client. They possibly have varying mobility from fixed users to walking or even driving a car. Also they could join and leave the network at any moment without prior notification. This dynamism of network topology should be respected when designing our autoconfiguration mechanism.

- Partitioning and merging: as indicated before, the ad hoc extension looses its reason to be if it's totally disconnected from the meshed backbone. Thus partitioning and merging constraints could be relaxed to cover only temporal disconnections. Nodes may switch from Mesh router to another as they move; this case should not be treated as a network partitioning or merging, rather simply as a case of topology change. For this end, Mesh routers should have a global vision of the ad hoc extension.

- Address limitations: in the EcoMesh context ad hoc nodes have to be able to access to the internet, thus used IP addresses must be globally available. Accordingly, the address range is limited; hence it must be carefully distributed and address leaks have to be detected and treated in a reasonable time.

- Energy and bandwidth constraints: collaboration between nodes is critical in the EcoMesh case. Intermediate nodes have to share their bandwidth and power resources to relay distant node's packets. The autoconfiguration mechanism should have limited communication needs.

- Reliable delivery: in the EcoMesh context like in normal ad hoc networks, the packet loss ratio is relatively high. The autoconfiguration mechanism should be flexible to overcome the unreliability problem.

3.4.2 Comparing the Existing Approaches

In this part we will compare the existing approaches to extract some conclusions and directions to better plan for our address autoconfiguration mechanism. First it could be interesting to note the state of advancement of this work (see table 3.1). It should be mentioned that the All DHCP Relay approach is the only approach that provides formal specification of the protocol as it inherits the DHCP specification, for the other papers only informal description is presented.

	Implementation	*Simulation*	*Modification at MAC layer*	*Approach*
Agent Based Addressing [3]	None	NS	Unspecified	Stateful
MANETconf. [4]	None	NS	Yes	Stateful
Prophet [5]	None	NS	Yes	Stateful
All Relay approach[7]	Yes	No	No	Stateful
Buddy Protocol [6]	None	NS	Yes	Stateful
SDAD [8]	None	None	No	Stateless
WDAD[9]	None	NS	No	Stateless
PDAD [10]	None	NS	No	Stateless
Ad hoc IP @ Autoconf. [11]	Work in Progress	None	No	Stateless
HCQA [12]	None	NS	No	Hybrid

Table 3-1: Underlying autoconfiguration approaches comparison

Second, we will compare the existing approaches based on a technical metrics that influence the design and the performance of the autoconfiguration mechanism and the whole ad hoc network (see table 3-2).

- Bandwidth consumption: this is one of the most important metrics; it also influences the power consumption. In the EcoMesh context, we try to convince nodes to accept this consumption by introducing incentive mechanisms. Since some protocols uses packets of variable size and requires two types of communications (periodic floods and per address assignment communications), we will divide the overhead into periodic flood and per address assignment overhead and computes it by number of packets per node.
- Latency: it is the time spent before configuring a node with a valid IP address. It's important to note that we assume here a reliable medium with zero loss.
- Sensitivity to network loss: network losses are inevitable in mobile ad hoc networks. Autoconfiguration protocols requiring long communications and excessive unicasts are the most sensitive to network losses. Higher sensitivity to network losses involves additional overhead and increased delays.

The table 3-2 illustrates the comparison between existing approaches based on the overhead, latency and sensitivity to network losses. We assume here zero packet loss. The following notation is adopted:

- N: total number of nodes
- d: the average network radius in number of hops.
- l: the average number of neighbors
- T: the period of synchronization, flood, or any repetitive procedure if exists
- k: the number of iteration if exists

– t: the round trip time for one hop communication

	Overhead per address assignment	*Periodic Flood*	*Latency*	*Sensitivity on network losses*
Agent Based Addressing	d packets per address assignment	Yes N packets per period T	T/2 + d*t/2	Very sensible
MANETconf.	2l + 2*N + N*d/2	No	(2 + d)*t	Very sensible
Prophet	2l packet exchange per address assignment	No	2*t	Not sensitive
All Relay approach	4*l*d	No	2*t*d + θ + ω See 3.5.2 and 5.2.3	Not sensitive
Buddy Protocol	2l packet exchange per address assignment	N^2 packets per period T	2*t (if the node has available IPs)	Not sensitive
SDAD	k*N	No	k*T (T is a timer)	Very sensitive
WDAD	overhead in routing protocol	No	Not an address assignment mechanism	Not sensitive
PDAD	No additional overhead	No	Not an address assignment mechanism	Not sensitive
Ad hoc IP @ Autoconf.	k*N + 128 bits per routing packet	No	k*T (T is a timer)	Very sensitive in the assignment phase
HCQA	k*N + d	Yes N packets per period T	k*T + T/2 + d*t/2	Very sensitive

Table 3-2: Performance comparison between existing autoconfiguration protocols

For example, if we take the Agent Based Addressing; it requires a request/reply communication with the Address Authority for each address assignment. If we consider a randomly placed node within the network; it will be on average "d/2" hops away from the AA. As a consequence the request/reply communication requires 2 packets relayed d/2 time each (d transmission). Also, this protocol requires the AA to flood the network periodically so N packets will be emitted. For the latency, each node have to wait for receiving a Verify packet from the AA before initiating its request, as an average it have to wait for T/2 time units given T the flood period then the request/reply communication will take d*t/2 time units because it's a communication between d/2 hops away nodes.

Finally, we compare the existing approaches based on evenness, routing dependency, distributed operation, address uniqueness and stability (see table 3-3).

– Address Evenness: As the addresses have to be globally available, this is an important metric in the case of EcoMesh. This metric gives an indication of the effectiveness of the address distribution. An even distribution implies a low address duplicate probability and better utilization of address space. For all existing autoconfiguration approaches, address evenness is achieved by design; the only exception is for the Buddy protocol. In this protocol, address assignment is accomplished by dividing the address rang between the requested and the requesting nodes. Thus if ad hoc nodes are concentrated in a particular zone within the network, they probably will run out of address availability while other nodes outside this zone have large address spaces. To overcome this

problem, the Buddy protocol implements a complex procedure to achieve address evenness by allowing requested nodes to ask for addresses within the network. The price will be more complexity and bandwidth consumption. In table 3.3, we will consider as "even" a protocol that achieves evenness by design and "uneven" a protocol that is either "uneven" or achieves evenness by additional measures.

- Dependency on routing protocol: in general, an approach dependent on specific routing protocol is better designed and should have better performance, but the advantage of an independent approach is its higher flexibility. In the EcoMesh context, if we are going to adopt a specific routing protocol, we should design the autoconfiguration mechanism to be compatible with this routing protocol and optimized for its characteristics.
- Distributed operation: in mobile ad hoc networks distributed operation is always preferred. The EcoMesh extension is characterized by its permanent connection to a stable backbone. Accordingly, we can tolerate centralization as the ad-hoc extension looses its reason to be if disconnected from the Mesh backbone.
- Address uniqueness: address duplicates may occur if two networks merge or in the address assignment phase with stateless approaches. In the EcoMesh context duplicates are not acceptable because other than the network perturbation, it may have a negative effect on the security or the incentive mechanism.
- Address stability: by address stability, we mean the possibility for unnecessary address changes. Address changes affect the stability of the network and lead to unnecessary overhead for assigning new addresses. In addition, all active communications will be corrupted when address changes leading to user's non satisfaction. Unnecessary address changes must be avoided.

	Evenness	***Routing dependency***	***Distributed operation***	***Address uniqueness***	***Address stability***
Agent Based Addressing	Yes	No	Centralized	Guaranteed	Low stability
MANETconf.	Yes	No	Distributed	Guaranteed	High stability
Prophet	Yes	No	Distributed	Not guaranteed	Not specified
All Relay Approach	Yes	No	Centralized	Guaranteed	High stability
Buddy Protocol	No	No	Distributed	Guaranteed	High stability
SDAD	Yes	No	Distributed	Not guaranteed	Not specified
WDAD	Yes	Yes	Distributed	Guaranteed with high probability	High stability
PDAD	Yes	Routing integrated	Distributed	Guaranteed with high probability	High stability
Ad hoc IP @ Autoconf.	Yes	Yes	Distributed	Guaranteed with high probability	High stability
HCQA	Yes	No	Semi-centralized	Guaranteed	High stability

Table 3-3: Characteristic comparison between existing autoconfiguration protocols

- Scalability: This metric is related to the communication overhead, the available address space and the address evenness. If the autoconfiguration mechanism requires excessive communications and periodic floods the mechanism won't be scalable, also if the address range is limited and the address distribution is uneven the mechanism will not scale well. In the EcoMesh context, the network is limited to a few hops and the address range will be limited, thus the address evenness will be of high importance in designing the mechanism. We should note that stateless and hybrid approaches are not suited for environments with limited address range.

3.5 Adopting the All Relay Approach

The All Relay approach is the most suited for the EcoMesh context, therefore we decided to adopt it as the address configuration mechanism for our hybrid network. The problem with all other approaches is that they suppose an isolated ad-hoc network that is not connected to the internet neither to an external network. Moreover, the stateless approaches do not guarantee address uniqueness even if they limit the duplicate probability. And all stateful approaches require an excessive traffic exchange to synchronize the allocation tables. In addition, all these approaches are not implemented or tested in a real environment. In contrast, the All Relay approach is suited for environments that require a globally available IP addresses and are connected to a fixed borne like in the EcoMesh case. On of the main advantages of this approach is its use of the wide accepted and largely tested DHCP protocol. In fact this approach does not require any changes on the existing protocol; rather it requires all nodes to run the relay service. In addition this approach fulfils all the EcoMesh requirements for address assignment.

- Topology change: the DHCP address lease time could be adjusted to correspond to the mean nodes lifetime. Nodes may leave the network on will and their assigned addresses will be reused after a maximum of "address lease" period also the departure detection need zero traffic. And after temporary disconnection, the node will keep using its address as long as the disconnection time does not exceed half "address lease" period. The DHCP guarantees the address uniqueness regardless of the topology change.

- Partitioning and merging: as indicated before, network partitioning and merging are relaxed to temporal disconnections in the EcoMesh context. After such a disconnection the node will keep using its IP address. When a node switches from a Mesh router to another, it will conserve its IP address as the DHCP server will not normally treat different Mesh routers as different subnets.

- Address limitations: the DHCP is characterized by an efficient use of the available address space. The node departure will be detected in a finite period of time with an upper bound of "address lease" period. When nodes lifetime within the network is very low, the lease period could be adjusted to correspond to this property. This modification change is transparent to ad-hoc users as it is on the server side.

- Energy and bandwidth constraints: the overhead introduced by the "All Relay" approach could be considered as acceptable seen that it requires no periodic flood, and the address request and address reallocation requires limited message exchange as their occurrence is separated by a relatively long period of time (in the order of "lease time"). The number of message needed per address allocation is studied later in this chapter.

- Reliable delivery: in the DHCP protocol, the address request is initiated by the client that is responsible for any retransmission after a packet loss. This guarantees the reliability of the protocol. Even more, the redundancy of the client/server exchange through many relays increases the reliability of the address assignment. The price is more bandwidth consumption.

The message exchange when requesting an IP address is the same as when using normal DHCP. In short, if the unassigned node "A" requests an IP address, the following steps will occur:

1. "A" broadcasts a DHCP-Discover message.
2. The neighboring relays of "A" intercept the message and forward it by unicast to the DHCP server.
3. The DHCP server replies by sending by unicast a DHCP-Offer to all the relays (those that forwarded the discover message).
4. The DHCP-Offer is then broadcast to "A" by its neighbors.
5. DHCP-Request / DHCP-Ack messages are exchanged the same way between "A" and the server. (now "A" is assigned with a valid IP address)
6. "A" starts the DHCP relay daemon to serve farther nodes to reach the access point.

3.5.1 Number of messages required per address allocation.

The number of messages required for assigning an IP address for a newly arrived node can be computed based on the known DHCP message sequence and according to the mean number of neighboring relays. If a node arrives at x hops from the Mesh born (we are only interested in the wireless overhead) and have k already assigned neighbors (k relays) then the average number of messages exchanged if all neighbors cooperate is in the order of:

$$messages\_exchanged = (4 * x * k)$$

This message exchange comprises a high redundancy; all neighboring relays that receive the client's DHCP broadcast will relay it toward the server. Thus the server will receives k duplicates of the same request and will reply by the same offer to all these relays. Accordingly, the redundancy rate introduced by this approach is about $\frac{(k-1)}{k}$.

Although the redundancy introduced is expansive in term of bandwidth consumption, it could be tolerated as long as it offers a simple way to reduce the sensitivity on network

loss; in addition, the amount of overhead introduced by this approach is less than the other autoconfiguration approaches especially that this approach does not require any network level flood and the period for address renewal is long enough compared with the periodic message exchange for all other approaches.

3.5.2 Delay before a DHCP address allocation

According to the DHCP RFC 2131, we can calculate the time needed for a client to get configured with a valid IP address. The figure 3.1 illustrates the timeline diagram of messages exchanged between DHCP client and server when allocating a new IP address.

After the DHCP client sends a DHCP-Discover it waits for a DHCP-Offer, if after a specified period of time it does not receive the offer the client retransmits the Discover message (the wait period is 4 seconds randomized by the value randomly chosen in the range of -1 to +1 secs for a 10Mb/sec Ethernet). This delay is long enough to allow sufficient time for replies from the server to be delivered to the client. Note that DHCP clients are responsible for all message retransmission. For subsequent retransmissions, a binary exponential backoff strategy must be adopted allowing for a maximum of 5 retransmissions.

This sequence of message exchange remains intact when the communication between the DHCP client and server passes through the intermediate of a DHCP Relay agent (when the client and the server are not on the same MAC broadcast). The only thing that changes is the number of hops taken by messages to reach their final destination.

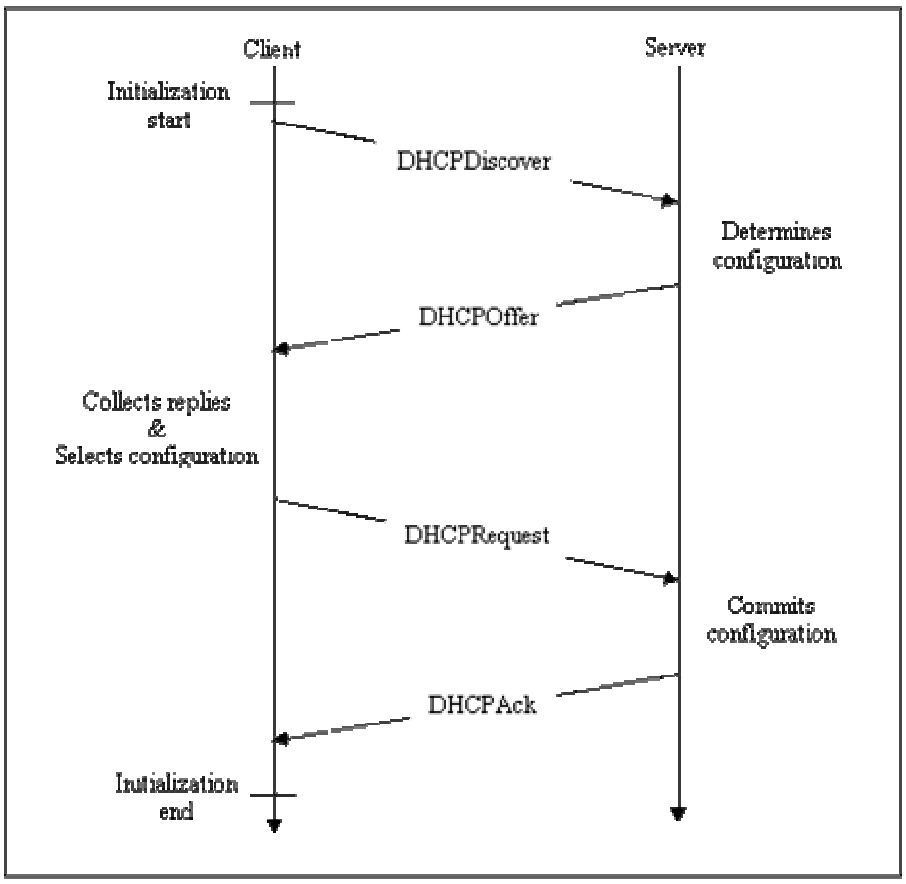


Figure 3.1: DHCP message exchange sequence

If we suppose a loss free communication, the assignment time is given by:

$$Assignment\_time = 4 * t * k + \theta + w$$

With:

- t: one way trip time for one hop communication.
- k: mean number of neighbors of a node.
- θ: complete address assignment treatment time by the server and the client.
- w: the client wait time for a response from the server.

We should note that the introduction of a wait time was left implementation dependent in the DHCP specification, so most implementations do not wait for responses after a client request rather the client chooses the first server who replies. For a comparison of address assignment time between real and virtual environments see 5.2.3.

3.5.3 Using the DHCP to limit the ad hoc range

The autoconfiguration protocol using DHCP relay agents could be used to assist in limiting the ad hoc range as required in the EcoMesh architecture. An ad hoc node is required to get a valid IP address before using the network. In the EcoMesh context the ad hoc extension will be limited for "K" hops maximum. This means that a node at

"K+1" hops away from the nearest Mesh router must not been allowed to use the network. Our described address configuration mechanism using DHCP relays could be used for this end. Ad hoc nodes that are at "maximum allowed hops" or simply at "K" hops from the nearest Mesh router should stop the DHCP relay module, and restart it only when they become at less than K hops from the Mesh.

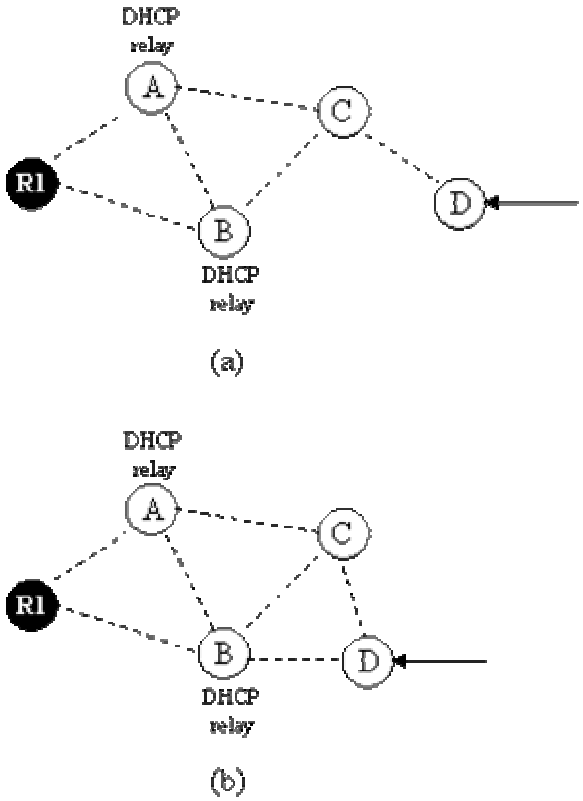


Figure 3.2: Limiting the ad hoc range using the DHCP relays

The example presented in figure 3.2 illustrates how limiting the ad hoc range could be achieved using the DHCP relays. If we suppose the network range limited to 2 hops and the ad hoc nodes at 2 hops away stop the DHCP relay module then in figure 3.2(a) when node D arrives, it tries to get an IP address by sending a DHCPDiscover message. Or node C that is at 2 hops can't relay the request as it does not run the DHCP relay, thus D will not be configured with a valid address. Therefore the range limitation could be accomplished by using the address configuration mechanism.
In the figure 3.2(b) D now has moved to the proximity of B that runs the DHCP relay. Now D could be assigned a valid IP as it is at 2 hops from the Mesh router. In addition, after been configured, D will not start the relay module as it is at 2 hops. The knowledge of the node's distance to the Mesh router is accomplished by the beaconing mechanism described in section 4.3.2.

In addition to assisting on limiting the network range, stopping the relay module at nodes that are "K" hops away reduces the bandwidth consumption for address assignment. For example, in figure 3.2(b) when D requests an IP only B will relay the request, C will not as it is at "K" hops away. This reduces the redundancy introduced by "All DHCP Relay" approach.

3.6 Summary

The problem of address configuration in the EcoMesh was the topic of this chapter. A simple mechanism that constitutes an extension to the DHCP protocol has been adopted. In hybrid networks, like the EcoMesh's one, a centralized DHCP server could be reached through the Mesh borne. The problem of broadcast communication has been solved by each ad hoc node running the DHCP relay agent module. This approach is the most suited for our case; its introduced overhead is limited and the assignment time is acceptable.

In addition to the adopted solution, the state of the art and the comparison study between existing autoconfiguration approaches in multi-hop networks is one of the contributions of this work as for I know, the presented study could be one of the most recent comparisons done on this topic.

Chapter 4

ROUTING AND MOBILITY IN ECOMESH

Most of existing routing protocols for ad-hoc networks are based on a flat architecture; many studies have proven the limitations of this flat architecture in term of control overhead and scalability. In [13, 14] they present that ad-hoc routing protocols work better under a limited ad-hoc network in term of number of hops.

In this chapter we will present our routing solution adapted for a hybrid wireless networks composed of a fixed stable Mesh backbone to which is attached limited range ad hoc networks. The particularity of this "EcoMesh" architecture calls for a routing protocol that better exploits the available infrastructure, global connectivity must also be supported as the network will be connected to the internet. Mobility support is also integrated in a part within the routing allowing for a seamless complexity free mobility support.

Following, a description of the requirements of the EcoMesh context in term of routing and mobility. Section 4.2 presents a background insight on related work. Section 4.3 uses the architecture described in chapter 2 to design the routing protocol taking advantage from the hybrid topology presented by the combination Mesh – adhoc. The three major components of our routing are explained; "Zone discovery" allows Mesh routers to discover nodes within their range and ad hoc nodes to register with the nearest mesh router, "Route discovery" allows for ad hoc nodes to connect to nodes within the Mesh network or on the internet and finally "Move detection and mobility support" allows for seamless mobility by exchanging some information between Mesh routers after a move has occur.

4.1 Requirements of the EcoMesh context

The particular architecture of the EcoMesh network presented in chapter 2 introduces some requirements on the routing and mobility.

- Limiting the ad-hoc horizon: As mentioned before, intra ad hoc communications are limited to few hops. Thus, to reach farther nodes the communication must pass through the Mesh backbone. The routing protocol must guarantee that direct ad hoc communications cannot occur between nodes that are more than the allowed hops away; rather, such a communication should only be possible by passing through the Mesh backbone. In addition the routing protocol must guarantee that nodes that are more than the allowed hop range away from the nearest Mesh router cannot use the services offered by the Mesh backbone.

- Building default routes: as the network will be used primarily to access to the internet, the most important requirement of routing is to maintain a valid path to the nearest Mesh router. Ad hoc nodes have to be able to identify the nearby nodes and should always choose the nearest one as a default gateway to communicate with nodes that are not in its local ad hoc range.

- Limiting control communications: the control communication overhead grows exponentially with the ad hoc network size. To build a scalable EcoMesh network, the control traffic should be limited. Network wide floods will not be acceptable. Accordingly the routing protocol must exploit the hybrid architecture to limit the control traffic (especially for route discovery) and it must guarantee that control messages will be limited to the zones involved in the communication.

- Mobility support: As wireless nodes are mobile, they must be able to move within the network without been disconnected; also, the mobility support should be transparent for end users. The mobility solution must guarantee that the Mesh backbone has a fresh and clear vision about the nodes location within the network.

4.2 Background and Related Work

Many papers have shown, by theoretical proof [15] or by real tests [16], the limitations in terms of capacity and scalability of pure flat ad hoc networks. The idea then, was of using the hybrid network architecture to better scale the network by combining the properties of ad-hoc and infrastructure based networks. This has been the topic of a lot of recent research activities. In this section, we will discuss some related work.

Early research with hybrid networks focused on the standard cellular systems. In [17, 18] the concept of adding ad hoc capabilities to existing cellular-based networks is studied. The ad hoc GSM (A-GSM) [17] aims to introduce relaying capabilities into the standard GSM network to decrease the occurrence of "dead spots" where services are not available. A user located in a dead spot could use the resources of another device, extended to support relaying, to reach the base station. To this end, each device broadcasts periodically a beacon. The beacon indicates if the node is directly connected to the base station, and the ID of the relay node through which the sender of the beacon can reach the base station along with the number of hops if it is not directly connected to the base station. This beaconing mechanism is used to select the best device or base station to communicate with. The Multihop Cellular Network (MCN) [18] presents an alternative to the traditional single hop cellular network by allowing for multi hops communications. Intra cell communications between devices can bypass the base station by relaying through other devices, thus allowing for simultaneous communications within a cell (over same channel) and therefore increasing the global throughput of the network by improving the spatial reuse of the system.

In [19] the interworking between ad hoc and cellular networks is studied. A "Mobile Gateway" that interconnects an IPv6 ad hoc network over a cellular network with the

6Bone was developed. The network can support the existence of multiple Mobile Gateways, and reactive and proactive Gateway Discovery mechanisms were considered to allow ad hoc nodes to find and set up a route to the nearest Mobile Gateway in order to be able to send packets to the Internet. Mobility in this approach is considered for a mobile node that moves between different heterogeneous networks (ex: from ad-hoc to company network). The solution is based on MobileIPv6 that provides seamless roaming.

The Zone Routing Protocol (ZRP) [20] combines reactive and proactive routing techniques. Each node maintains fresh local routing information about its surrounding topology (local zone) by using a proactive routing scheme, the Intrazone routing protocol (IARP), and reactively discovers routes to distant nodes when needed, the Interzone Routing Protocol (IERP). The search throughout the network is done by efficiently querying selected nodes in the network (bordercasting) as opposed to querying all nodes like in traditional ad hoc reactive routing protocols. The network division is made on node's level, so routing zones of neighboring nodes overlap. The nodes local zone is defined by the *zone radius*, the number of hops allowed to treat a node as local. This parameter could be adjusted according to the mobility level in the network, the more the nodes are mobile the better to use lower zone radius, and the more the nodes are fixed the better to use higher zone radius. Comparatively, our architecture uses the Mesh routers rather than ad-hoc nodes to accomplish the Interzone communications. Moreover ad hoc nodes in our architecture should proactively maintain only the path for the nearest mesh router. Proactively maintaining routing information about surrounding nodes is not considered relevant as the intra ad hoc communications are minor.

The Lightweight Underlay Network Ad-Hoc Routing (LUNAR) [21] protocol implements a layer in between the MAC layer and the IP layer to perform a variation of multi-hop ARP. The main idea behind it is to link ad hoc path establishment to ARP. The protocol is simple and is designed to work in 3-hop environments. LUNAR uses a combination of reactivity and proactivity for route discovery and maintenance. It is reactive in the sense that it discovers paths only when required; it is proactive in the sense that it rebuilds active routes from scratch every 3 seconds, even if everything is fine with the current path. This removes the need for additional path maintenance procedures and link repair actions.

The Landmark Ad Hoc Routing (LANMAR) [22] applies the clustering paradigm to scale routing in flat ad hoc networks. LANMAR assumes that the large scale ad hoc network is grouped into "logical subnets", reflected by a consistent addressing scheme, and in which the members have a commonality of interests and are likely to move as a "group". LANMAR uses the notion of landmarks to keep track of such logical subnets. Each logical group has one node, dynamically elected, serving as "landmark". The route to a landmark is propagated throughout the network using a Distance Vector mechanism. For local operation, each node runs the Fisheye State Routing (FSR) with a scope range of a defined maximum hop distance. As a result, each node has detailed topology information about nodes within its "Fisheye scope" and has a distance and routing vector to all landmarks. When a node needs to relay a packet, if the destination is within its Fisheye scope, accurate routing information is available from the Fisheye Routing Tables. Otherwise, the packet will be routed towards the corresponding landmark based on the destination address. However, if the packet arrives within the scope of the destination

before reaching the landmark, it is routed to it directly without going through the landmark. Thus, the LANMAR scheme largely reduces the routing table size and the routing update traffic overhead. Comparatively, the Mesh nodes in the EcoMesh network do not rely on addressing schemes to define zones; rather they rely on the ad hoc scope (maximum allowed hop distance). In addition, the concept of mobile groups in LANMAR does not hold in the EcoMesh context where Mesh routers are fixed and serve mobile ad hoc nodes. Finally, ad hoc nodes in the EcoMesh context maintain routing information only to the nearest Mesh router; in contrast, in addition to local routing information, ad hoc nodes in LANMAR hold routing information to all landmarks within the network.

In [23] a hierarchical ad hoc network structure is proposed by using the LANMAR protocol and a clustering technique. This approach assumes that among the mobile nodes, there exist "backbone nodes" (BNs) that have an additional powerful radio to establish wireless links between themselves forming a higher level network. The LANMAR that is logically structured is extended to profit from the physical hierarchy created by BNs. Thus, packets destined to a node outside the local zone of a node will be routed to the nearest BN that will direct it to the corresponding BN based on the address information. This will reduce the number of hops. This approach differs from ours in its dependence on address schemes to define groups, although that it defines a mechanism, inherited from LANMAR, to allow for a node to move to a different zone (difference here is based on addressing) but the mechanism is complicated. Second, our approach makes no use of a clustering algorithm, and the Mesh routers, equivalent to "Backbone nodes", are fixed and considered as always available making no need for an election algorithm.

A hybrid version of AODV called Hierarchical AODV (H-AODV) is presented in [24]. Like in [23], the network comports "Backbone nodes" equipped with additional powerful wireless interface and forming what is called the "backbone network". Mobile ad hoc nodes are grouped into clusters comprising a dynamically elected BN as the cluster head each. A mobile node will select the nearest BN within its "K-hop scope" (defining the cluster range) to be its cluster head. In H-AODV, basically both backbone nodes and ordinary nodes run the same AODV routing protocol. However, the backbone nodes will broadcast RREQ packets throughout the backbone network in addition. By recording the interface ID for each route learned, later RREP packets and data packets can also utilize the backbone links. The key point of H-AODV is that the route discovery procedure can take the "short cut" advantage of the physical hierarchy. Thus, the backbone links are usually utilized to route packets to remote destinations. H-AODV uses an additional field in the routing control packets to define subnets around each BN. This field stores the backbone node address of the source node, which issues this control packet. Each node only re-broadcasts routing packets, which contain the same backbone node address as itself. Thus, all packets destined to outside the local subnet must pass through the associated BN. This is not true in our architecture where nodes are able to directly communicate with any node within the maximal hop bound even if the corresponding node is registered with another Mesh router.

4.3 The Proposed EcoMesh Routing Protocol

Our routing protocol attempts to take advantage of a combination of proactive and

reactive components like in the ZRP protocol in order to fulfill the specific requirements of the EcoMesh context described in 4.1. The proactive part consists of each Mesh router to discover the ad hoc nodes within its range and of each ad hoc node to register and build a route with the nearest Mesh router. The reactive component is to discover and maintain routes to nodes involved in an active communication either they are source, destination, intermediate ad hoc nodes or they are Mesh routers. The described protocol is based on the famous AODV protocol and consists of the following entities:

- Zone discovery: the process by which Mesh routers discover nodes within their zone and by which, ad hoc nodes discover and build a route to the nearest Mesh router. This constitutes the proactive part of our routing protocol.

- Route discovery: the mechanism by which ad hoc nodes discover a route to a destination, as the destination is another ad hoc node within the network or an internet destination. This part constitutes the reactive part of the protocol.

- Move detection and mobility support: the mechanism used to detect nodes mobility. This mechanism is integrated with routing as move detection in multi-hop environment will take place on the network level.

- Inter-Mesh communications: this is the mechanism that specifies how Mesh routers will handle the communications.

The protocol is slightly different for the Mesh routers than normal ad hoc nodes. For the Mesh routers, more functionalities are needed to take profit from the Mesh backbone and to handle the attached nodes. In the following sections we describe in detail the entities of our proposed protocol and the related implementation issues based on the AODV protocol.

4.3.1 AODV overview

The Ad-Hoc On-Demand Distance Vector (AODV) [25] routing protocol uses a packet exchange to establish routes. A source node wishing to communicate with a destination node broadcasts a Route Request (RREQ) packet and expects to receive a Route Reply (RREP) either from the destination or from an intermediate node with a fresh route to the destination. The RREP is unicast back to the source. Upon receiving the RREP, the source can begin sending data using the path that has been set up during the route discovery process.
Routes generated using the route discovery process are temporary and expire after some time if not recently used. If source's route to the destination is deleted and there is additional data to send, the route discovery process is initiated again. Nodes forwarding RREQs and RREPs use the route information contained in the packets to learn routes to other nodes. These routes are stored temporarily in a cache and often have a shorter lifetime than the routes stored at the source and destination nodes. To avoid processing old packets, each broadcast packet is uniquely identified by the node's IP and the Broadcast ID maintained by each node and incremented with each initiated RREQ. Sequence numbers are also used to determine the freshness of routes.
AODV uses Route Error (RERR) packets to signal nodes of unreachable destinations.

When an active link breaks, the node upstream of the broken link informs its upstream neighbors by issuing a RERR. The RERR is propagated by each upstream node depending on its own cached routes. Nodes that receive a RERR may decide to initiate route discovery for that destination if a route is still desired.

In contrast to all other reactive based ad hoc routing protocols, The AODV was raised in 2003 to full RFC status. In addition, for AODV, formal validations have been carried out by Verinet group [26]. Using a theorem prover and a SPIN model of AODV in a 2 node setup, it was shown that it is in fact a loop free routing protocol[1].

4.3.2 Zone discovery

Each Mesh router in our protocol has to be aware of the ad hoc nodes in its zone. To this end, our routing protocol employs a Mesh router driven beaconing mechanism. This beaconing process is performed at the network layer to support the multi hop environment. The beaconing mechanism allows ad hoc nodes to learn about Mesh routers within K hops scope (K represents the maximum allowed ad hoc range, in the rest of this chapter K will be used for simplification). The ad hoc nodes are involved in a registration process by which they choose the best Mesh router to use as a default gateway for communicating with destinations beyond the K hops distance. Thus the zone discovery phase in our protocol could be divided into two complementary sub processes; the beaconing process and the registration process.

Beaconing process

Periodically, each Mesh router transmits a BEACON packet. The BEACON is an extended RREP packet with “destination IP address” and “originator IP address” fields set to the IP address of the wireless interface of the Mesh router. The BEACON is broadcasted within the network with a TTL of K allowing all ad hoc nodes within K hops to know about the presence of the router. The beacon also contains a BEACON_INTERVAL which indicates the period of beacon transmission. Finally, each Mesh router increments its sequence number and includes it in the beacon upon transmission. The sequence number field allows ad hoc nodes to detect fresh beacons as they maintain the most recent sequence number received and therefore beaconing is loop free (it is the same sequence number behavior of standard AODV).

Upon receipt of a BEACON, the ad hoc node checks if a registration process is required (explained later in this part). If the beacon is from the Mesh router with whom the node is registered, the node will add/update a routing entry in its routing table to the originating Mesh router. The route entry is set as a default route and set to expire after a timeout period based on the number of allowed beacon losses and the beacon interval included in the beacon packet. The ad hoc node will use this route entry for its communications with nodes more than K hops away. Non duplicate beacons are scheduled for retransmission if the TTL field is equal to 1 (before being decremented). The “hop count” field is incremented prior to retransmission to account for the current hop.

[1] Even if validating the protocol on a 2 nodes setup may be insufficient (does not cover the entire sequence tree) it remains between the rare formal validations of ad hoc routing protocols.

Registration process

At any moment, a connected ad hoc node must be registered with only one Mesh router. The ad hoc node should always choose the best Mesh router to register with. In our protocol, the criterion of choosing the best Mesh candidate is the number of hops. Even if this could not be the best criterion as it does not take into account the possible congestion either on intermediate nodes or on the Mesh router, we believe that such criterion is acceptable in a hop limited scenario.
The registration process is initiated by the ad hoc node by sending a REGISTER message directly to the corresponding Mesh router (in unicast) using the route already built by the beacon message so no route discovery is needed. The REGISTER packet is an extended RREP packet. The ad hoc node transmits a register message if any of the following conditions are met:

- Condition 1: Upon receipt of a beacon and the node is not registered with any Mesh router.

- Condition 2: Upon the receipt of a beacon from the corresponding Mesh router and the node is involved in an active communication through the Mesh router.

- Condition 3: Upon receipt of a beacon from a nearer Mesh router.

- Condition 4: Periodically, in case the node has no active sessions through the Mesh router.

The first condition exists when a node join the network or become connected after a long disconnection period. In this case the node is also not registered with any other Mesh router. The registration procedure must be reliable; after the REGISTER message is sent, the node expects to receive an acknowledgment from the Mesh router. This could be done simply by setting the "A" flag (Acknowledgment) in the register message (as it is a RREP packet). If after a REGISTER_DELAY period the node does not receive an Acknowledgment from the Mesh router, it reinitiates the registration procedure by sending a REGISTER packet.

When the node is involved in communication through the Mesh router, a valid route should always be maintained at the node toward the Mesh router (through periodic beacons) and at the Mesh router toward the ad hoc node. For this end, when an ad hoc node receives a beacon from the corresponding Mesh router and has active sessions, it responds by a register message. This message has double role; first it updates the routing entry toward the node at the Mesh and the intermediate nodes, second it updates the registration state of the node at the Mesh router. In this case, it is not necessary to send back an acknowledgment as the registration will be periodic thus can tolerate the occurrence of a loss (the values of the parameters is explained later in this chapter).

Ad hoc nodes should always choose the nearest Mesh router to register with. Thus, when a node receives a beacon from a nearer Mesh router (identified by the hop count field) it should send a register message to the nearer Mesh router. The register message should contain the IP of the Mesh router with whom the node is currently registered; this

information is important as we will see later to allow for a seamless mobility. This information could be included in the register (RREP) as an extension. During this transition, the node keep using the old Mesh router as a default gateway, and starts using the new one only after it receives the acknowledgment to the register message, then it deletes the routing entry toward the old Mesh router and sets the entry of the new one as a default gateway. If an ad hoc node receives a BEACON from a nearer Mesh router while it is involved in a registration process with another Mesh router, it should reregister with the new discovered router, as each ad hoc node should register with the nearest router, and include the IP of the farther router to insure the unique registration constraint.

When the node has no active sessions through the Mesh router, responding to each received beacon could be relaxed. In this case, our protocol requires each ad hoc node to send periodically (REGISTER_PERIOD) register message to the Mesh router to refresh its registration. Reliability in this case is necessary so the Mesh router has to reply by an acknowledgment. This lazy registration process, in case there is no active routes, permits to consume less bandwidth resources for control packets. This is reasonable too in the sense that it is not necessary for Mesh routers to maintain routes to all nodes in their zones. Only routes to active nodes should be maintained. If after a REGISTER_DELAY period the node does not receive an Acknowledgment from the Mesh router, it reinitiates the registration procedure by sending a REGISTER packet.

The REGISTER messages are used by each Mesh router to build the “registration table”, a table used to identify all nodes within a Mesh router’s zone with their distance in number of hops. Each entry in the table is associated with a timer. At timeout, the Mesh router deletes the corresponding entry from the table. When the Mesh router receives a new registration message it adds/updates the entry in the table. When the node detects that the node has move to another Mesh zone, it deletes the entry from its table (move detection explained later in this chapter).

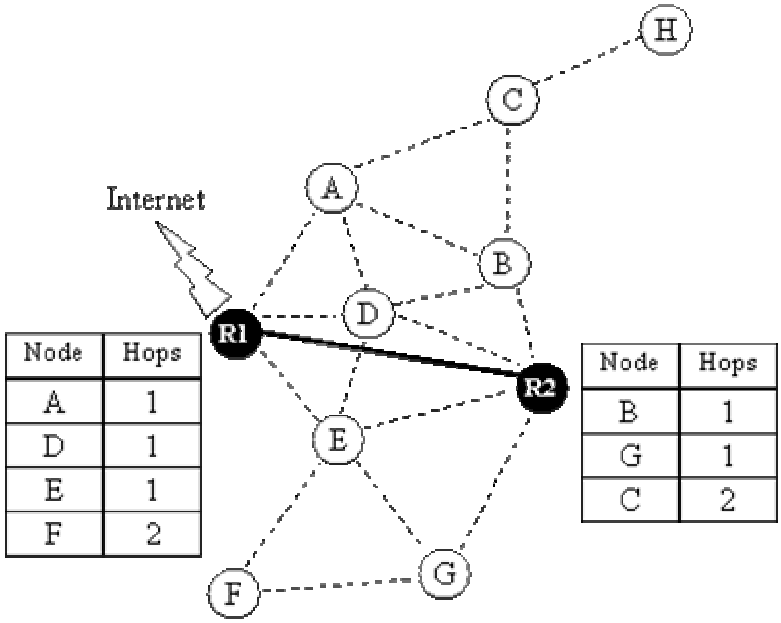


Figure 4.1: Registration Tables for simplified Internet connected EcoMesh network

As an example, the registration tables for Mesh nodes are represented in the figure 4.1, the network comports 2 Mesh routers and 8 ad hoc nodes. One of the Mesh routers is connected to the internet. If we consider the zone range limited to 2 hops, node H will not be included in any of the registration tables as it is 3 hops away from both Mesh routers. For node D, even if it is at 1 hop from both nodes, it is only registered with R1 (same for nodes C, E and F).

4.3.3 Route discovery

Route discovery is the phase concerning the determination of a valid route to a destination when an ad hoc node has application layer data to send. The route discovery procedure is different for destinations that are within the mesh network than those for Internet destinations.

Connection to a node within the Mesh network:

Nodes within the Mesh network are identified based on their subnet ID; as IP addresses are distributed through DHCP, they share the same subnet. When an ad hoc node has data packets to send to a node within the Mesh, it checks its routing table for a valid entry to the destination. The route discovery process begins when the node has no such a valid entry.

In response to this event, the node initiates a RREQ packet with a TTL of K. The RREQ is then broadcasted within the network and includes the destination being searched for and the originator's IP address (as specified in AODV). The sequence number handling is the same as for the AODV protocol.

When an intermediate ad hoc node receives the RREQ, it checks if the destination IP is its own IP address; if this condition is met, it replies by unicasting a RREP to the originator node. The RREP will take the path created by the RREQ packet. If the intermediate node is not the corresponding destination IP, it checks the TTL value and rebroadcasts the packet if the TTL is not 0 (after decrementing the TTL). In our protocol, intermediate ad hoc nodes must not respond to RREQ on behalf of the destination node even if they have a fresh valid entry to the destination. We believe that due to the limited broadcast range, replying to RREQ by intermediate nodes is not as important. Even more, if we take the network represented in figure 4.1, imagine node A has an open session with node G through the Mesh router R1. Now if node C has data to send to node G, it initiates a RREQ to this destination, normally the route should be C-B-R2-G. If intermediate node A responds to the RREQ initiated by C, the connection will pass through R1 which is not the nearest Mesh router to C (C is registered with R2), this violates the requirements of the protocol that requires to use the nearest Mesh router (the one with whom the node is registered). That's why we decided to disallow intermediate nodes to reply on behalf of destination nodes.

When a Mesh router receives a RREQ it reacts in a different way. It uses the registration table and its prior knowledge about all Mesh nodes within the network to assist it to better respond to the RREQ (this assumption is realistic as Mesh routers are fix and the Mesh topology is stable, in addition it belongs to the same operator as specified in the

EcoMesh description in chapter 2). The Mesh router responds in a different way if the RREQ is received from an ad hoc node or if it is received from another Mesh router. the Mesh router can detect if the RREQ is received from another Mesh router if the source IP matches one of the IPs of known Mesh routers. Alternatively, the Mesh router can detect if the RREQ is received from an ad hoc node or a Mesh router by looking to the interface over which it was received.

When a Mesh router receives a RREQ from an ad hoc node it does the following:

- If the ad hoc node is not a registered node, the RREQ is simply dropped as no route should pass through the Mesh router for an unregistered node. Thus for a given RREQ, multiple Mesh routers will not try to process it.

- If the ad hoc node is a registered node, the Mesh router is responsible of processing the request. For this end, if the Mesh router has a fresh enough routing entry to the destination, it will reply on behalf of the destination. If no such entry is available, it looks first if the destination IP in the RREQ is a registered node. If yes, it rebroadcasts the request using its wireless interface after resetting the TTL field to K; this insures that for a route discovery between two nodes within the same Mesh router's zone the RREQ will reach all the nodes in the zone. When the destination IP is not a registered node the RREQ will be sent to all other Mesh routers over the backbone link.

When a Mesh router receives a RREQ from another Mesh router (over the backbone link), it drop the packet if the destination IP is not a registered node, and rebroadcasts the packet over its wireless interface after resetting the TTL field to K otherwise. This insures that RREQ broadcasts will only affect the zones involved in the communication and thus reducing the routing control overhead.

RREP generated by destination nodes or the corresponding Mesh router, are then unicast back to the originator taking the shortest path in number of hops created by the RREQ.

Internet connectivity

When an ad hoc node has data to send to the internet, i.e. the destination's subnet mask does not match that of the ad hoc network, the node uses its default route. The default route is built as explained in 4.3.2. When an intermediate node receives a data packet for an internet destination, it forwards it using its default route and adds an entry in its routing table pointing to the reverse path toward the packet source node.; this allows constructing a reverse path toward the source node without the need for RREQ/RREP procedure. Thus for a connection to the internet, the only requirement is to have a valid default route. Recall that a default route is a synonymous to the route to the nearest Mesh router and is built through the registration procedure. Now when a Mesh router receives a data packet originating from the internet and destined to an ad hoc node, it checks it routing table for a route to the destination. If a route does not exist, a route discovery procedure is performed for the destination (as explained in the previous part). Our protocol allows to external nodes to initiate connections to the internal ad hoc nodes and in contrast to several propositions that combines Mobile IP with the ad hoc routing

protocol, or uses IP in IP encapsulation to connect to the internet our protocol only requires some modifications on the routing protocol without the need to integrate it with any additional component.

4.3.4 Move detection and mobility support

In the EcoMesh context, like any wireless environment, ad hoc nodes are mobile and may arbitrary move within the network. With node move, the location information must be updated to reflect the current position of the node and thus its point of attachment. In EcoMesh, the location of a node is synonymous to the Mesh router with whom the node is registered.

As explained earlier, one of the goals of the registration process is to supply (1) Mesh routers with ad hoc nodes that are within their zone, (2) and ad hoc nodes with the nearest Mesh router to register with. Accordingly, our mobility solution will be based on the registration process and integrated in some part within the routing protocol. Ad hoc nodes are responsible of detecting their mobility and of informing the Mesh side about their location by registering with the nearest Mesh router. On the other side, a Mesh router "R1" is responsible of maintaining the list of all nodes that are in its own zone and of informing the Mesh router "R2" when an ad hoc node moves from the R2's zone into R1's zone. The requirement we impose onto our mobility solution is that at any moment an ad hoc node is always registered with only one Mesh router.

When an ad hoc node moves away from the Mesh router with whom it is registered, it begins to receive BEACON messages from other Mesh routers, and it decides to register with a new Mesh router only when it receives a beacon message with a hop count less than that of the router with whom it is currently registered. At this moment, the node sends a REGISTER message to the new router including the IP of the current Mesh router, and expects to receive an Acknowledgment to confirm the success of the registration. During this new registration process, the node continues to use the old Mesh router as a default gateway as long as a valid route exists to this router and no Acknowledgment is yet received. When the node receives the Acknowledgment message, it deletes the route to the old Mesh router and adds one to the new Mesh router. At this moment, the node considers that the mobility has occurred successfully.

On the Mesh level, when a Mesh router receives a REGISTER message containing the IP of the actual Mesh router of the node, it concludes that a move has occurred. The Mesh router is responsible then to inform the old Mesh router about the move; and all other Mesh routers to modify their forwarding tables to point to the new location of the node if they have active routes to the ad hoc node in question. Next we explain the message exchange on the Mesh level after move detection.

- The new Mesh router sends a MOVE_NOTIFY message that includes the IP of the old Mesh router, the IP of the new Mesh router and the IP of the ad hoc node "all Mesh routers". Sending the packet to all Mesh routers could be done by a simple broadcast if the Mesh routers are on the same LAN or through a multicast address that group all Mesh routers if the Mesh routers are not reached by a simple broadcasting mechanism. In all cases, how the Mesh routers are

connected is not a part of our study; our protocol requires only that the Mesh routers could be reached by a certain mechanism. After sending the MOVE_NOTIFY packet, the Mesh router expects to receive an acknowledgment from the "old Mesh router" within a MOVE_NOTIFY_DELAY period of time. At timeout the Mesh router retransmits the MOVE_NOTIFY packet destined to the "old mesh router" IP.

- When the "old node" receives the MOVE_NOTIFY packet, it deletes the entry corresponding to the ad hoc node from its registration table. Thus the requirement that an ad hoc node is only registered with one Mesh router at a given time is respected. Then, it sends back an acknowledgment to the "new Mesh router".
- When a Mesh router other than the old router receives the MOVE_NOTIFY packet, it updates its forwarding table to point to the new Mesh router if it has active routes with the indicated ad hoc node. If no route exists, the Mesh router simply discards the packet.
- When the new Mesh router receives the Acknowledgment from the old node, it initiates an acknowledgment back to the ad hoc node indicating that the procedure was successful.

This message exchange is represented in figure 4.2.

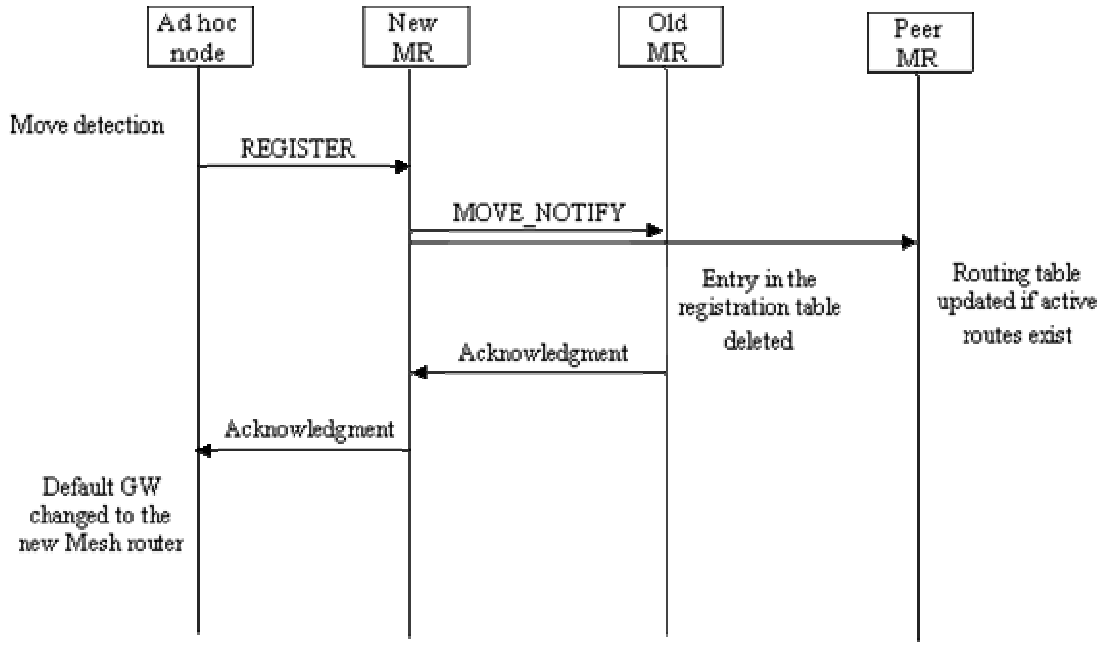


Figure 4.2: Message exchange after move detection

4.3.5 Inter-Mesh communications

As stated in chapter 2, our work in this project was first limited to the ad hoc extension. But with the evolvement of the study we discovered that this limitation was not realistic as there will be some information exchange on the Mesh level that affects the ad hoc

operation. For example, the Mesh routers need to pass the route requests on the Mesh backbone to discover distant nodes. Message exchanges are required on the Mesh level to support nodes mobility…

On the Mesh backbone, the stability of the routers and their non mobility could be exploited by more efficient routing protocols. Proactive routing could also be used as the Mesh topology is stable. Even more, on the Mesh level, ad hoc routing protocol may not be the best to be used. In this part we will describe the requirements on Mesh routing to support the ad hoc operation, we are not suggesting what routing protocol to use on the Mesh level; for instance any existing routing protocol could be used as long it integrates the requirements stated below.

Our first requirement is that "all Mesh routers" could be reached by some mean. This could be through a simple broadcast if the Mesh routers share the same broadcast network, or through a multicast address that groups all the Mesh routers.

As mentioned in the "Route Discovery" section, when a Mesh router receives from an ad hoc node a RREQ destined to a node that is not listed in its registration table, it should retransmit the RREQ to all Mesh routers on the backbone interface. The Mesh routing protocol have to be able to differentiate between this packet and other normal routing packets. The following treatment is required: when a Mesh router receives a RREQ on its backbone link, it should retransmit the request on the wireless interface only if the destination IP is included in its registration table; the packet is dropped otherwise.

When a Mesh router receives a packet from the internet destined to a node within the Mesh network (identified by the destination IP), the Mesh router should buffer all received packets and initiate a route discovery procedure. It must send a request to all Mesh routers as if it has received a RREQ from another Mesh router. If after a specified period of time no reply is received it drops the buffered packets and sends back to the source an ICMP destination unreachable message.

4.3.6 AODV modification and implementation issues

As stated earlier in this chapter, our proposed routing protocol as described in this report is based on the AODV specification. The EcoMesh protocol requires some modifications on the normal AODV behavior at the same time specific parameters should be considered. In this part we will describe the AODV modifications and implementation issues.

BEACON Packet

In our protocol the beacon message is an extended RREP packet with the destination IP and originator IP fields set to the IP address of the Mesh router initiating the Packet. In addition, this RREP should contain the following extension defining its type as BEACON.

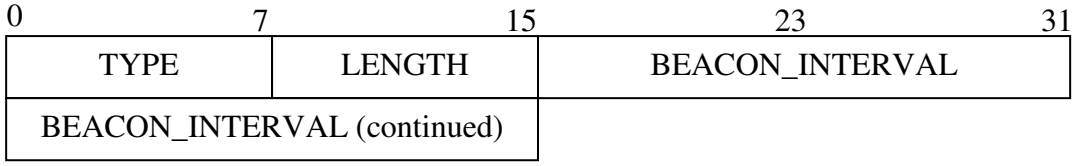


Figure 4.3: BEACON extension format

With:

Type: BEACON packet
Length: 4 (the Type and Length are not included)
BEACON_INTERVAL: the time in milliseconds between two successive beacon packets.

We should node here that the BEACON extension has the same format of the HELLO format defined in AODV, the only difference is the TYPE that must be different to allow ad hoc nodes to treat the packet as a Beacon and not normal HELLO message that could be initiated by any node.

BEACON_INTERVAL

For the inter-beacon period a compromise between network overhead and performance consistency should be considered. A smaller value means higher overhead but at the same time better accuracy in term of mobility detection delay and route refresh. In the EcoMesh case we propose to use the value of 1500ms.

REGISTER Packet

The REGISTER packet is an extended RREP packet with the originator IP that of he Mesh router to register with, and the destination IP that of the ad hoc node initiating the REGISTER message. This message comports an extension to allow the Mesh router and intermediate nodes to treat the message accordingly. Note: if the REGISTER do not contain the defined extension, it will be dropped by intermediate nodes as the AODV protocol specify to drop each RREP that is not the response of a previously received RREQ.

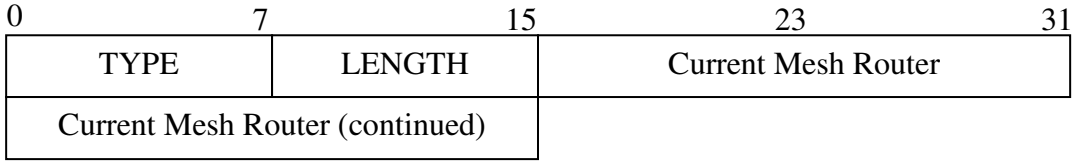


Figure 4.4: REGISTER extension format

With:

Type: REGISTER packet
Length: 4 (the Type and Length are not included)
Current Mesh Router: indicates the Current Mesh router with whom the node is currently registered. It is set to “all zeros” if the node is not registered with any other node.

In Many cases, the registration procedure should be reliable; the ad hoc node expects to receive an acknowledgment after sending the REGISTER message. As REGISTER are extended RREP messages, reliability could be achieved by simply setting the "A" flag in the message. When the ad hoc node has an active session through the Mesh router and needs to send a REGISTER message, the "A" flag will not be set; in all other cases the "A" flag must be set.

REGISTER_PERIOD

When the ad hoc node has no active sessions through the Mesh backbone, it initiates a REGITER message with a period of REGISTER_PERIOD. The proposed value of this timer is relaxed to a value 64 seconds. This high value is allowed as when the node is inactive there is no need to refresh the route toward it at the Mesh router. This allows conserving bandwidth to data packets. The timer is initiated after a successful registration; it is stopped after a connection through the Mesh router, and reinitiated when no active sessions exist.

REGISTER_DELAY

When the ad hoc node initiates a reliable registration process, it expects to receive an Acknowledgment to confirm the registration. Thus, it starts the REGISTER_DELAY timer, and reinitiates the registration process at timeout. The value of REGISTER_DELAY must be calculated as follows:

$$2 * NODE\_TRAVERSAL\_TIME * (Hop\_Distance + TIMEOUT\_BUFFER)$$

With Hop_Distance the separation between the ad hoc node and the Mesh router in hops.
Note: NODE_TRAVERSAL_TIME and TIMEOUT_BUFFER are defined in section 10 of the AODV RFC.

Default route

Normally AODV treat the same way destination nodes as they are ad hoc nodes or internet destinations. In fact, AODV, like all ad hoc routing protocols, is designed for isolated ad hoc networks in essence. To support internet connectivity, we have defined default routes; when a node has data packets to send it checks the IP of the destination and forward packets to the default gateway if the destination is on a different subnet. This should be integrated within the AODV implementation so RREQs will be only initiated for "local destinations" and default gateway used only for "remote destinations".

4.4 Summary

In this chapter we have presented a simple routing protocol suited for hybrid EcoMesh like networks. The proposed protocol takes advantage of the existing Mesh backbone to limit the broadcast nature of route discovery procedure. At the same time the Mesh utilization is avoided when direct ad hoc connection is present. The protocol combines reactive and proactive components; reactivity is allowed on the ad hoc part to discover ad hoc nodes within the Mesh network; while proactivity concerns Mesh routers to discover

the nodes within their zone and to announce their presence to allow ad hoc nodes to discover reachable Mesh routers. Each ad hoc node maintains a default route to the nearest discovered Mesh router so routing to the internet is possible by using proactively the available default route. Nodes mobility is supported within the routing protocol and by introducing some data exchange on the Mesh level. Each node is required to register with the nearest Mesh router so nodes location is known at the Mesh level. Upon moving from zone to zone the involved Mesh router informs the others about the "handover" occurring to make the changes if needed (update routing entry to point to the new location, delete an invalid entry in the registration table…).

Work is still ongoing to introduce the required modifications into the AODV routing protocol. The protocol will be different for ad hoc nodes and Mesh routers that require additional functionalities.

Chapter 5

TEST – SIMULATION

In this chapter we explain some tests we made to study the connectivity in a hybrid network. And we explain our emulated environment that will serve for testing the protocol (after completing the implementation). This chapter will be divided into two parts: (1) testing the network connectivity, (2) building the simulation environment.

Section 5.1 presents the connectivity tests devised to study (1) the node's connectivity in function of the cooperation level, network size and density, (2) the relaying charge on cooperative nodes. The EcoMesh architecture could be exploited when taking the appropriate network density and inter Mesh separation. Section 5.2 presents the emulation environment we are working on. It is constituted on Linux virtual machines and a wireless network emulator. Using this environment allow us to test the modifications we are working on in a relatively real environment and on real machines. This allows us to pass directly to real implementation.

5.1 Part 1: Testing the network connectivity

In this part we explain the tests we made to test the variation of network connectivity in a hybrid network with the variation of node's cooperation level, network density and size. The tests were completed using a C program that we wrote for this end. The goal of these tests is to allow us to draw the limits of hybrid networks from the dimensional point of view.

5.1.1 Background comparison

The problem of network connectivity was considered in several papers, the main research issue was to find the number of neighbors needed to consider the network as connected. in a recent paper [27] they show that the number of neighbors needs to grow like $\Theta \log n$ in order to consider the network as connected, where n is the number of nodes randomly distributed (uniformly iid) in a unit square. However the above analysis considered only cooperative ad hoc nodes. For the EcoMesh context, where nodes may not cooperate and where connection is considered as the ability to reach the Meshed backbone, these results do not hold. For this purpose we make several tests in order to study the connectivity rate in function of the network density, network size and the cooperation level.

One could say that the connectivity problem could be solved by increasing the network density (as consequence the number of neighbors). Though a dense network will potentially be highly connected, the network capacity will decrease. Suppose node i

wants to send a message to node j, all j's neighbors wont be able to transmit nor to receive, and the message transmission will increase interference at i's neighbors. This comes from the broadcast nature of wireless communications where nodes have to share the medium according to the IEEE 802.11 MAC specification [28]. In [15] Gupta and Kumar proved that for identical randomly located nodes of fixed bit rate and fixed connection range, the throughput obtainable by each node for a randomly chosen destination is proportional to $\frac{W}{\sqrt{n \log n}}$ under noninterference protocol, Where W is the node bit-rate and n the number of nodes.

The particularity of the EcoMesh like network is that in a part, it allows for nodes to not to cooperate and in another part, since the probability to meet the same users is very low, the communications between ad-hoc nodes and the Meshed backbone will be dominant (intra ad-hoc communication is always possible but is minor). The latter implies that nodes have to share the effective bandwidth of the meshed access point to which they are attached (in an ad-hoc manner). As a result the obtainable throughput for each node will decrease as the distance between mesh routers increase and as the network density increases for a given inter mesh routers distance. Thus the tradeoff between network connectivity and mean per node capacity should be carefully examined.
Therefore, we started our tests in order to find the acceptable network density and size that first give acceptable connectivity even with high non cooperating nodes, and second conserves acceptable per node capacity.

5.1.2 Test model

A simple way to analyze the EcoMesh network was to consider it as a square area with optimally placed mesh routers as illustrated in figure 5.1; thus the whole network may be seen as the assembling of smaller squares comporting each only one mesh router and its attached ad-hoc network, the “basic network”. At the time of writing this report, the simulated network was limited to one basic network constituted of a number of randomly distributed (uniformly iid) nodes. Extending the model to include 4 Mesh routers is an ongoing work, but the test results are not available yet.

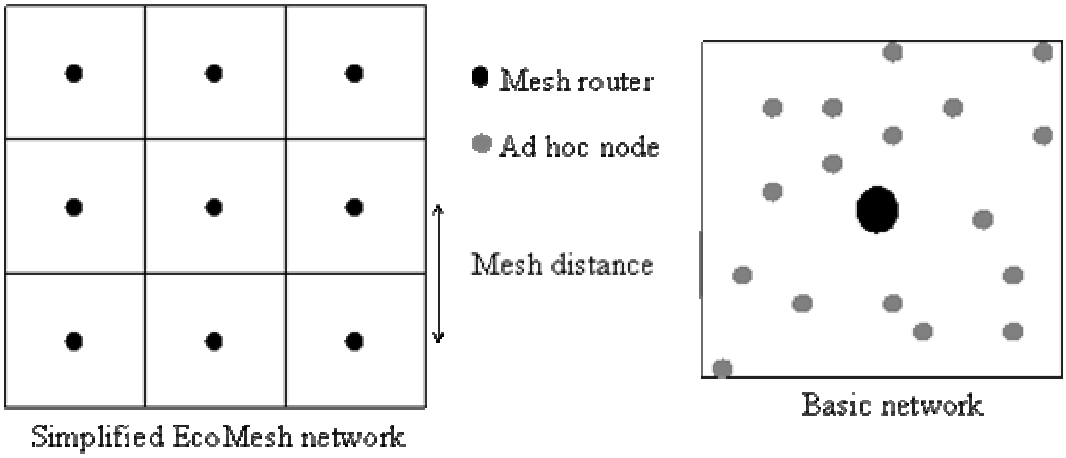


Figure 5.1: Simplified analyzed network model

The connection range of a wireless node was considered as (*d*); the Mesh distance is represented in multiple of node's connection range (a Mesh distance of *3d* means that the distance between two Mesh routers or simply the leg of one basic network is equivalent to 3 times the wireless connection reach). Finally, the density of the network is represented in term of nodes per *d* square (n/d^2).

5.1.3 Results

Many tests were devised in order to study the network connectivity and the relaying charge with the variation of the network density and size and with the variation of the percentage of cooperating nodes. The connectivity in our tests is equivalent to the existence of a path between a node and the Mesh router. Thus, after randomly distributing the nodes, the test program constructs the connectivity graph of the network constituted of the shortest path between all nodes and the Mesh router. Accordingly the network connectivity graph could be represented as a tree with the Mesh router as the root.

Before looking at the results, it is worth noting that the data presented here particularly in term of network connectivity is an overstatement of the real connectivity that could be reached on real environment. The reason comes from the fact that in our tests the data exchange, the bandwidth share and even the behavior of the routing protocol were not considered. The only consideration was the existence of a path between an ad hoc node and the Mesh router. As such, the results are instead comparative and serve as an indicator of the degree of connectivity of the network and the capacity share between nodes.

The results are obtained by repeating each test 10000 times to totally tolerate the randomized distribution and to give accurate results.

The effects of cooperation density and size on network connectivity

In this series of tests we aim to study the variation of network connectivity with the cooperation level, network density and size.

First, the disconnection rate was studied in function of the density and network size when all nodes cooperate. The node density was varied from 1 to 10 nodes per d^2 in steps of 0.25, and the Mesh distance was varied from 2d to 6d in steps of 0.2. In figure 5.2 we can observe that as the density increases the disconnection rate decreases, 3 behaviors should be noted (look at the left plot of figure 5.2):

- High density $>3/d^2$: for a node density greater than $3/d^2$ the disconnection rate is less than ~1% and about 0% for density greater than $4/d^2$ despite of the network size. This result is logical as for a high network density, nodes have a large number of neighbors (8.04 neighbors for a density of $3/d^2$ and a network length of 3d and 10.32 neighbors for a density of $4/d^2$ and a network length of 3d) thus have a better probability to reach the Mesh router even if the network is large.
- Moderate density ($2/d^2$ - $3/d^2$): for a reasonable density, the network connectivity still relatively high especially with a network size less than 3d (92.6% for a density of $2/d^2$ and a network size of 3d). We will focus more on

this density range in the further tests.

- Low density $<2/d^2$: for a network density less than $2/d^2$, the disconnection rate increases dramatically (it is 29.4% for a network density of $1/d^2$ and 16.6% for a density of $1.5/d^2$ and a network length of 3d for both densities). In addition the disconnection rate in this density range increases rapidly with the network size (it is 43.0% for a network density of $1/d^2$ and 22.3% for a density of $1.5/d^2$ and a network length of 4d).

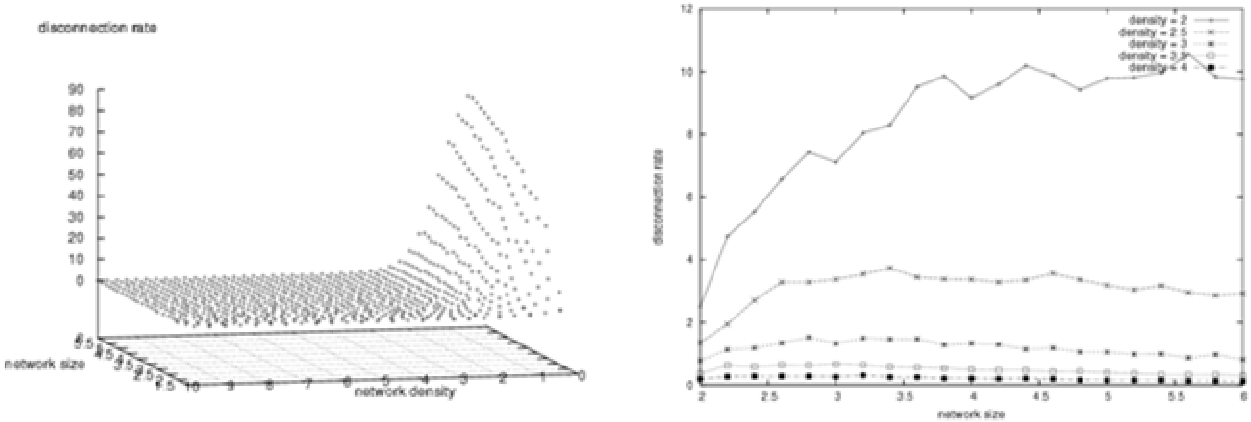


Figure 5.2: Disconnection rate in function of network density and size

Then we started to study the disconnection rate when introducing non cooperating nodes. A non cooperation node is a node that is always a leaf in the connectivity tree; it does not allow for other nodes to use it as a relay. The cooperation rate was varied from 100% to 40% in a steps of 5%; first it was considered with a fixed network density of $3/d^2$ in order to observe the variation of the disconnection rate with the network size (figure 5.3); then the cooperation rate was considered with a fixed network length of 3d in order to observe the variation of the disconnection rate with the network density (figure 5.4).

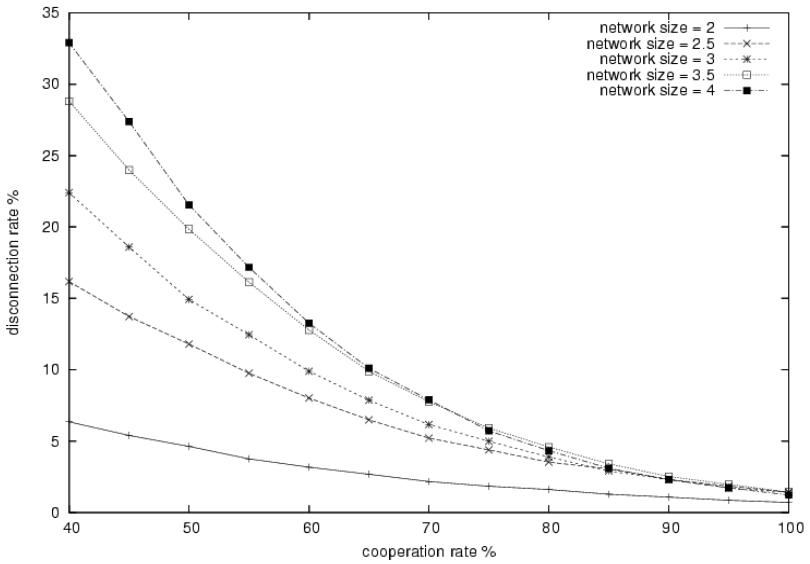


Figure 5.3: Disconnection rate in function of cooperation level and network size

In figure 5.3 we can observe the variation of disconnection rate with the cooperation level for different network sizes at a density of $3/d^2$. We notice that for a cooperation level more than 70% the connectivity remains high for all network sizes (up to 90.3% for a size of 4d). The connectivity then decreases rapidly for cooperation less than 70%. For a fixed cooperation level, the disconnection rate increases with the network size, this is logical as with higher network size more nodes are distant from the Mesh router and therefore have less chance to reach the Mesh. Even if the cooperation is very low, the disconnection rate remains under an acceptable level for a realistic network size (22.1% for a network size of 3d and 15.9% for a network size of 2.5d and a cooperation rate of 40% for both).

In figure 5.4 we can observe the variation of disconnection rate with the cooperation level for different network densities and a fixed size of 3d. It is clear that the disconnection rate increases with the decrease of network density and cooperation, but this time too, it remains acceptable for a realistic density such $3/d^2$.

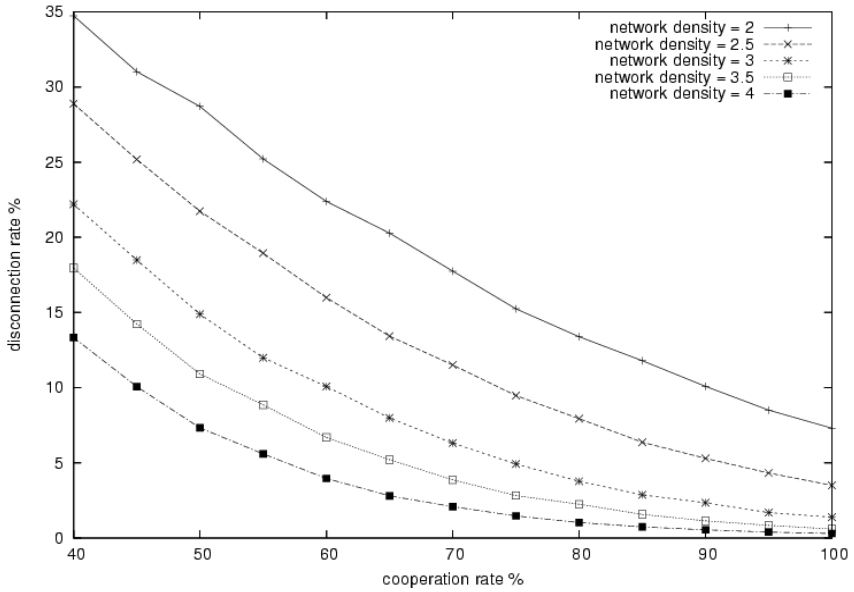


Figure 5.4: Disconnection rate in function of cooperation level and network density

The effects of cooperation level, network density and size on the relaying charge

In this series of tests we aim to study the variation of the relay charge with the cooperation level, network size and two specific densities, $2.5/d^2$ and $3/d^2$.
This study is important as each ad hoc node in the EcoMesh network will potentially serve as a relay for farther nodes; therefore have to share its available bandwidth. Thus, it is important to answer the following question:

To how many nodes a closer node has to relay the packets?

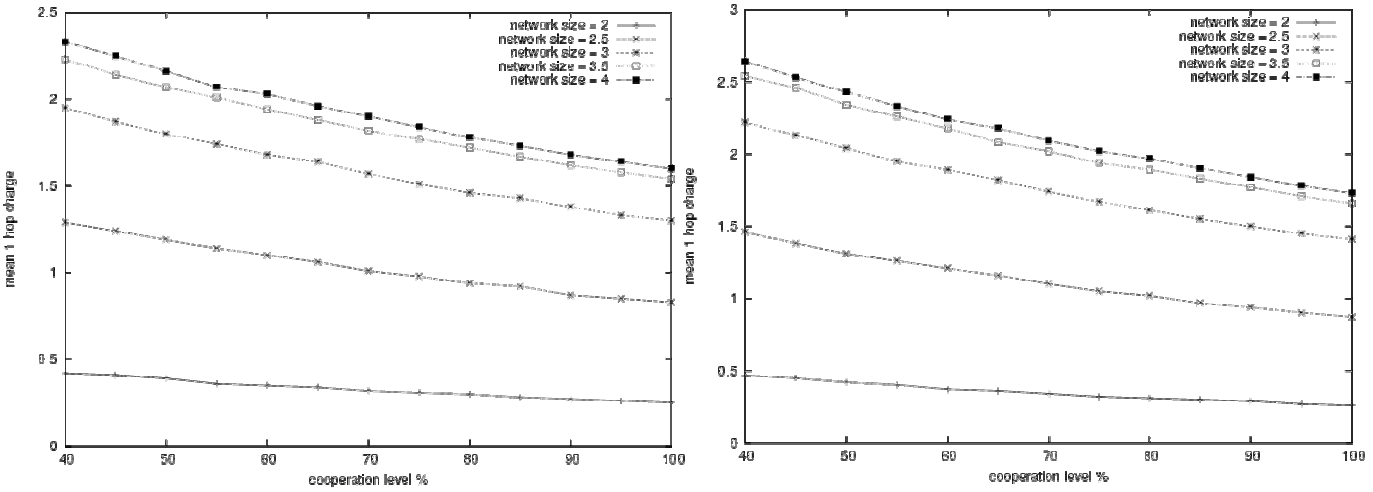
This metric (the relaying charge) gains additional importance when we consider cooperating and non cooperating nodes. Only cooperative nodes will act as relays thus will probably suffer from increasing charge when the cooperation rate decreases. The figures 5.5 – 5.7 illustrate this relaying charge. The (a) plots in these figures correspond to a density of 2.5 and the (b) to a density of 3.

Figure 5.5 illustrates the relaying charge on nodes that are one hop away from the Mesh borne. By relaying charge we mean the rate of direct children (nodes at 2 hops away) only (in other words: how many children have a cooperating node at one hop away). The first observation is that the relay charge does not increase significantly even when the cooperation reach only 40%. This is explained by the fact that when cooperation decrease, (1) the disconnection rate increases (one hop away nodes are always connected) (2) routes to the Mesh borne become non optimal leading to additional hops to reach the Mesh. Thus cooperative nodes will not pay highly their cooperation. Another observation is that for both densities the charge is very close, it is slightly higher with a density of $3/d^2$.
Per example for a density of $3/d^2$ and a size of 3d the charge is about 1.44 for a full cooperation (in other words the proportion of 2hops away nodes over 1hop away node is 1.44) and 2.21 for a cooperation of 40% and it is 1.31 for full cooperation and 1.95 for a cooperation of 40% for a density of 2.5 and the same network size.

The same observations hold for the figure 5.6 where the relaying charge of two hops away nodes (proportion of 3 hops away nodes over cooperative 2 hops away nodes) is represented in function of cooperation level. We can also notice that for a network dimension of 3d, the proportion of 3hops away nodes is relatively low for both densities.

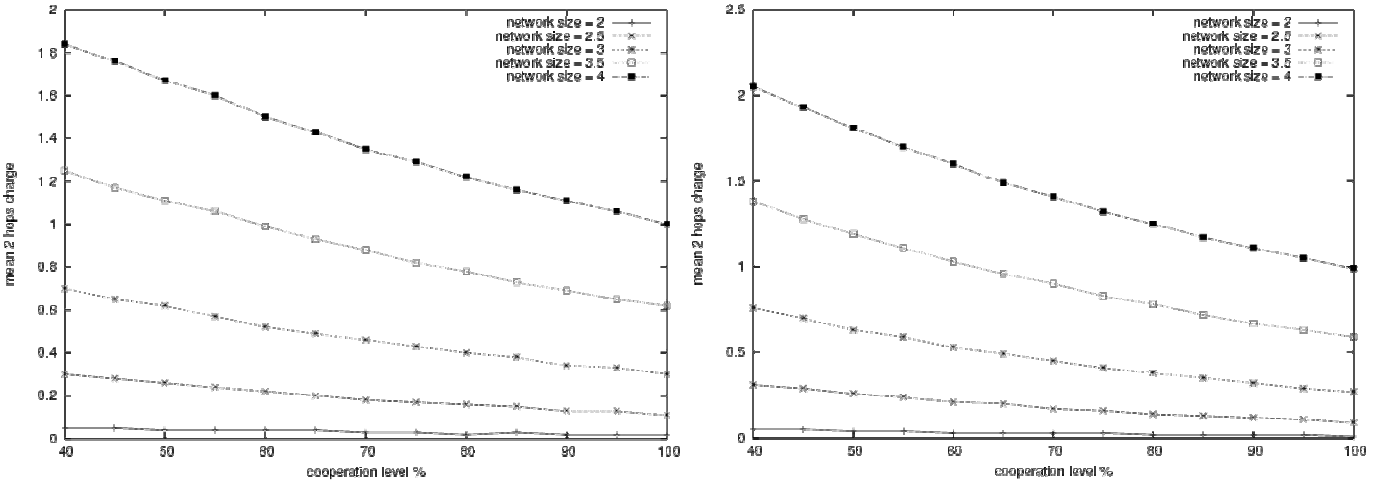
The last figure 5.7 represents the relay charge over two hops away nodes, with the difference that it considers all nodes more than 3 hops away as children. When comparing figure 5.6 and 5.7 it is clear that for a network size of 3d and 2.5d the proportion of 4 and more hops away nodes is negligible. This is of high importance in the EcoMesh context where ad hoc range limitation is essential. With these inter-mesh distance the network architecture itself will aid to guarantee this hop limitation as there will be always (with very high probability) a Mesh router less than 4 hops away.

According to these results we can deduce that for a moderate density in the range "2.5 – $3/d^2$" and an inter-mesh distance in the range of "2.5 – 3d" the network will still highly connected for a cooperation rate greater than 70%, even when the cooperation decreases to 40% the connectivity still highly acceptable especially for the $3/d^2$ density (21.1% et 15.9% with a network size of 3d and 2.5 d respectively). On the relaying charge, one concern was the possible high increase of relaying charge when cooperation decreases. It was shown that this was not the case; the relaying charge will increase slightly when cooperation decrease thus cooperative nodes will not to pay an extra charge by relaying to more nodes. The last result states that if we will be placed in the 2.5 – 3 density range and 2.5 – 3 inter-mesh distance range, the allowed hop limit could be fixed to 3.



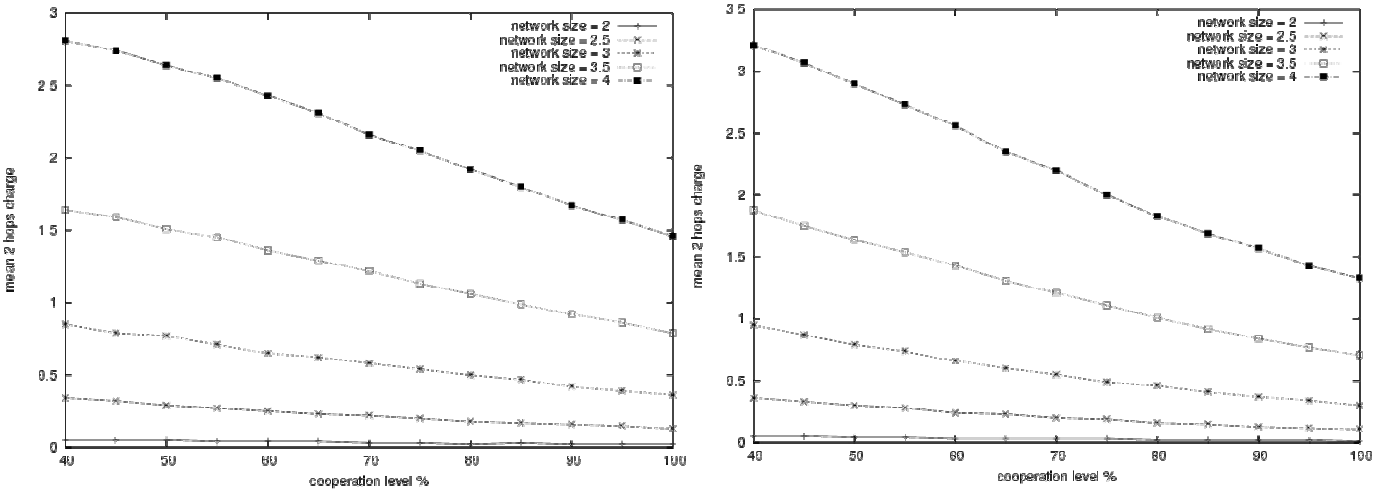
(a) (b)

Figure 5.5: Relaying charge on 1 hop away nodes



(a) (b)

Figure 5.6: Relaying charge on 2 hops away nodes (with nodes only 3 hops away)



(a) (b)

Figure 5.7: Relaying charge on 2 hops away nodes (with nodes 3 ore more hops away)

5.2 Part 2: building the simulation environment

In this part we present the work we have done in order to build our "emulated" environment. The goal of this work was to test in a reasonably realistic environment the behavior of different components that constitutes the EcoMesh solution. The first choice was to use, like in most of network studies, a network simulator like NS or GLOMOSIM. The problem of using such simulation tools is that they do not give accurate results especially in a wireless environment; even more the work should be done twice if we are looking to implement our solution. For this, we decided to take another untraditional way and use LINUX virtual machines along with an emulation infrastructure for multi hop wireless communications. Next we will explain in short the components of the test environment and some tests we have done.

5.2.1 Simulation framework

Our simulation framework is constituted of Linux virtual machines, virtualized wireless driver and the environment emulator.

Linux virtual machines

Running many virtualized Linux machines on a single host machine is possible with the implementation of User Mode Linux (UML). With UML, the Linux kernel is ported to the user space. A Linux virtual machine emulates the whole operating system and allows it to run independently from the host machine. The advantage of using Linux virtual machines is that they allow us to run any already existing tool under Linux without any modification. A virtual network could be also built using virtual machines.

Virtual wireless driver and environment emulator

The architecture of WIFI-for-UML [29, 30], the wireless emulator we are using, is represented in the figure 5.8. As illustrated it is composed of a virtualized wireless driver (hostap) used by the UML virtual machines and a simulation server that integrates the virtual wireless cards and a physical layer emulator.

The virtual wireless driver (hostap) was modified to run on top of the UML machines. A simple bus implementation was needed to export the required symbols for the hostap driver to be inserted into the UML kernel (the UML does not contain any bus implementation). The bus (named netbus) is implemented as a client TCP. The hostap driver act on a virtual device integrated in the emulator. The virtual device reacts, from the driver perspective, as a real hardware. The connection between the emulated wireless card and the driver is realized with a TCP connection, thus it is possible to run multiple UML machines on different hosts to distribute the simulation charge. When the driver is inserted in the UML kernel, it initiates a TCP connection to the emulator that links the driver with a virtual device. Then a virtual machine is created in the emulated environment. The machine could be fixed or mobile according to a mobility strategy implemented within the emulator. The basic emulator was supplied with a simple random mobility; this was extended in [31] to implement more controlled mobility strategy. The physical layer emulator uses a threshold based propagation model. Packets transmitted

between two machines are delivered only if the distance between the machines is less than the reception range. Even if more complicated and realistic propagation models could be added, we are using the emulator with the supplied propagation model.

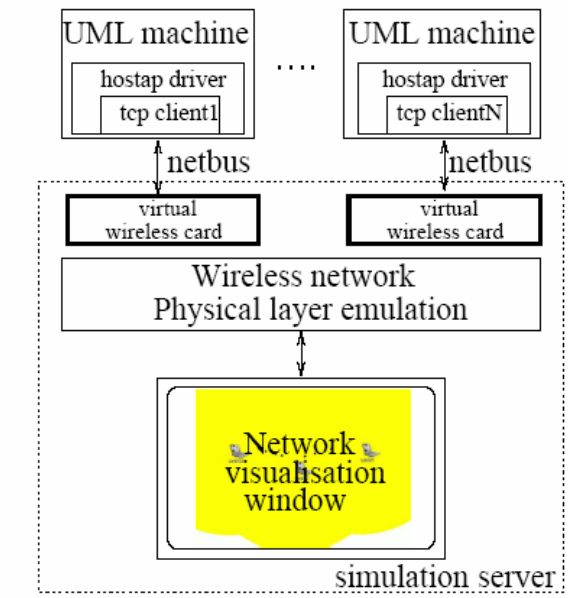


Figure 5.8: Architecture of the used wireless emulator

Building the simulation environment involves mainly building the UML machines. This is divided into two tasks, building the UML core kernel and building the UML file system. This was a time consuming task as the installation of different modules and the required packages (DHCP, FTP, libpcap, AODV…) on the virtual machines requires some modification to the default installation procedure so it is not always straightforward. In [32] detailed steps are presented in order to build the virtual machines. After building the simulation environments, it was possible to test any modification we made in a relatively realistic environment. As an example, the AODV modifications we are working on (work in progress) could be directly tested on the emulator and on a real machines too which allows us to pass directly between simulation and real implementation.

5.2.2 Adapting the AODV routing protocol

The AODV routing protocol constitutes the base of our routing solution. Modifications to the normal behavior of AODV are needed to meet our solution requirements and some characteristics of the EcoMesh context. In particular, allowing nodes to not playing the role of relay is possible in our context and was implemented directly in AODV.

Implementing the cooperation notion

All AODV implementations consider only cooperative nodes; this behavior doesn't hold in our situation, the users will be able to choose whether to cooperate or not according to different service fees, energy level, network usage...

For this end, we have modified the behavior of the RREQ process so that the ad hoc node treats only the RREQ destined to its own address when it is in a “non cooperation” state, and of course it treats all received RREQs when it is in a “cooperation” state (normal AODV behavior). When ad hoc nodes act as relays for farther nodes they consume more energy and bandwidth. Or ad hoc nodes are usually mobile thus battery powered, their energy level will work an important role in the cooperation decision. In addition the service fee the user is paying plays also an important role. In order to make the cooperation decision more realistic and dynamically adjusted, we simulated the cooperation state by considering the battery level and the service fee (we considered two different fees Regular and Premium); we integrated the cooperation notion into the routing protocol. The result was then a routing protocol that periodically checks if its resources allows it to cooperate and behaves accordingly. So, ad hoc nodes are now able to switch between cooperation and non cooperation states according to local decisions while using the network.

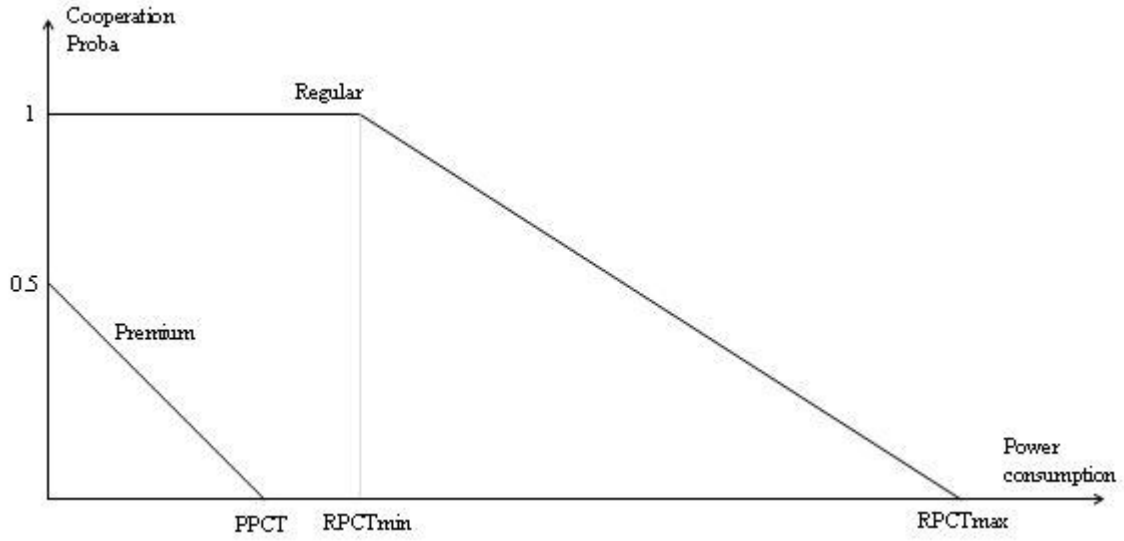


Figure 5.9: Cooperation state in function of power consumed and service type

5.2.3 Comparing DHCP assignment time

We used the built emulation environment and a testbed used by the project to compare the DHCP allocation time of our adopted solution described in section 3.5. The testbed used is constituted of two laptops running under Linux equipped with wireless interfaces and 3 PDA running under Familiar distribution of Linux. One of the Laptops runs the DHCP server and the other DHCP relay. The DHCP client is by default installed on the PDAs. The considered network is constituted of linearly distributed nodes as illustrated in figure 5.10. Table 5.1 presents the obtained mean assignment time (test repeated 50 times); we should note that routing between intermediate nodes was done statically.

Figure 5.10: Tested network for DHCP assignment time

	Emulation environment	Testbed
1 hop	1049.6ms	538ms
2 hops	1145.5ms	549.2ms
3 hops	1202.2ms	567ms
4 hops	1250.75ms	590.25ms
5 hops	1301.6ms	-
6 hops	1367.35ms	-

Table 5-1: DHCP assignment time comparison between real and virtual environments

The assignment time presented in table 5.1 show that for additional hops the assignment time increases slightly; this additional time corresponds only to the time needed to pass an additional hop. Thus the treatment time could be considered as constant. We should note also that the DHCP clients used on virtual and real machines are different and both do not implement a wait time after the DHCPDiscover message.

5.3 Summary

In the first part of this chapter we presented some tests in order to study the connectivity and charge in the EcoMesh like hybrid context. According to the obtained results, it is possible to conserve a good connectivity even when cooperation is low with a moderate network density and inter-mesh distance. In addition, one concern was the possible high increase of relaying charge when cooperation decreases. It was shown that this was not the case; the relaying charge will increase slightly when cooperation decrease thus cooperative nodes will not suffer from an extra charge.

In the second part of this chapter, we explained the emulation environment we are working on. Even if it was hard and time consuming task to build the emulated environment based on Linux virtual machines, this environment allows us now to test any modification we make in a reasonably realistic environment and to pass directly to real implementation thus gaining time.

Chapter 6

CONCLUSION AND FUTURE WORK

The purpose of this project was to propose a solution that is tailored for the EcoMesh hybrid network architecture and requirements described in chapter 2. The key idea is to extend the Meshed hot-zone by a collaborative ad hoc extension.

A mechanism based on the DHCP protocol for address assignment was adopted. Each arriving ad hoc node runs the DHCP relay after getting a valid IP through DHCP. This circumvents the problem of broadcast communications required by DHCP. This approach is totally suited for the case of hybrid networks like the EcoMesh one where a centralized DHCP server could always be reached.

A modified routing protocol based on AODV was proposed. It takes advantage of the existing Mesh backbone to limit the broadcast nature of route discovery. The Mesh routers are used to communicate with distant nodes and to access to the internet. Direct ad hoc connection is possible between nodes separated by only few hops. The protocol combines reactive and proactive components; reactivity is allowed on the ad hoc part to discover nodes within the Mesh network; while proactivity is achieved through a beaconing and registration mechanisms; Mesh routers transmit periodically BEACON messages to allow ad hoc nodes to discover reachable Mesh routers and build default routes; ad hoc nodes register with the nearest discovered Mesh router and use it for distant and internet connection. Each router holds the list of registered ad hoc nodes.
Nodes mobility is supported within the routing protocol and by introducing minor data exchange on the Mesh level. Nodes location is known at the mesh level through the registration mechanism. Upon moving from zone to zone the involved Mesh router informs the others about the “handover” occurrence to make the required changes (update routing entry to point to the new node’s location, delete an invalid entry in the registration table).

Work is still ongoing to introduce the required modifications into the AODV routing protocol. The usage of a “light AODV” on the Mesh side is also a work to do; this reduces the complexity of the required interaction between two different routing on the ad hoc and mesh side.

The connectivity of this architecture was tested by considering one Mesh router and its attached ad hoc extension. Results show that allowing multi-hops provides a high gain in term of covered zone; the network still highly connected with a moderate cooperation level and network density. It was also shown that relaying charge is not highly affected by the cooperation level for a moderate density. In these tests, the model studied was very simple; extending the tests to a more complex network model is a work to do.

Future work also includes performing simulation tests using the built test environment that includes mobile nodes with real data transfer in order to test the performance especially the throughput with an unmodified routing and using our proposed routing. We believe that our solution will improve the global throughput of the network.

Bibliography

[1] R. Droms, “Dynamic Host Configuration Protocol”, Network Working Group, IETF RFC 2131, March 1997.

[2] S. Cheshire, B. Aboba and E. Guttman, “Dynamic Configuration of IPv4 Link-Local Addresses”, Network Working Group, IETF RFC 3927, March 2005.

[3] M. Günes and J. Reibel, “An IP Address Configuration Algorithm for Zeroconf Mobile Multihop Ad Hoc Networks,” *Proc. Int’l. Wksp. Broadband Wireless Ad Hoc Networks and Services*, Sophia Antipolis, France, Sept. 2002.

[4] S. Nesargi and R. Prakash, “MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network,” *Proc. IEEE INFOCOM 2002*, New York, NY, June 2002.

[5] H. Zhou, L. M. Ni, and M. W. Mutka, “Prophet Address Allocation for Large Scale Manets,” *Proc. IEEE INFOCOM 2003*, San Francisco, CA, Mar. 2003.

[6] M. Mohsin and R. Prakash, “IP Address Assignment in a Mobile Ad Hoc Network,” *Proc. IEEE MILCOM 2002*, Anaheim, CA, Oct. 2002.

[7] Matthew J. Miller, William D. List, and Nitin H. Vaidya, “A hybrid network implementation to extend infrastructure reach”, Technical report, University of Illinois, Champaign-Urbana, January 2003.

[8] C. Perkins Charles Perkins, Jari Malinen, Ryuji Wakikawa, Yuan Sun and Elizabeth M. Belding-Royer, “IP Address Autoconfiguration for Ad Hoc Networks,” IETF draft, 2001.

[9] N. H. Vaidya, “Weak Duplicate Address Detection in Mobile Ad Hoc Networks,” *Proc. ACM MobiHoc 2002*, Lausanne, Switzerland, June 2002, pp. 206–16.

[10] K. Weniger, “Passive Duplicate Address Detection in Mobile Ad Hoc Networks,” *Proc. IEEE WCNC 2003*, New Orleans, LA, Mar. 2003.

[11] J. Jeong, J. Park, H. Kim, H. Jeong and D. Kim, “Ad Hoc IP Address Autoconfiguration,” IETF draft, August 2005 (work in progress).

[12] Y. Sun and E. M. Belding-Royer, “Dynamic Address Configuration in Mobile Ad Hoc Networks,” UCSB tech. rep. 2003-11, Santa Barbara, CA, June 2003.

[13] Xiaoyan Hong, Kaixin Xu and Mario Gerla, “Scalable routing protocols for mobile ad hoc networks” IEEE Network, pages 11 – 21, July 2002.

[14] Atsushi Iwata, Ching-Chuan Chiang, Guangyu Pei, Mario Gerla and Tsu-Wei Chen, "Scalable routing strategies for ad hoc wireless networks", IEEE Journal on Selected Areas in Communications, 17(8):1369 – 79, August 1999.

[15] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks", In IEEE Transactions on Information Theory, vol IT-46, no. 2, PP. 388-404, March 2000.

[16] P. Gupta, R. Gray, and P. R. Kumar, "An Experimental Scaling Law for Ad Hoc Networks," May 16, 2001.
http://black1.csl.uiuc.edu/~prkumar/

[17] G. Aggelou and R. Tafazolli, "On the relaying capacity of next-generation GSM cellular networks", IEEE Personal Communications, 8(1):40 – 47, February 2001.

[18] Y. Lin and Y. Hsu "Multihop cellular: A new architecture for wireless communications", In Proceedings of the IEEE Conference on Computer Communications (INFOCOM), Tel Aviv, Israel, March 2000.

[19] Nico Bayer, Dmitry Sivchenko, Bangnan Xu, Sven Hischke, Veselin Rakocevic and Joachim Habermann, "Integration of Heterogeneous Ad hoc Networks with the Internet", International Workshop on Wireless Ad-hoc Networks (IWWAN) 2005, London UK, May 2005.

[20] Zygmunt J. Haas, Marc R. Pearlman and Prince Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks", IETF draft, July 2002 (work in progress).

[21] Christian Tschudin and Richard Gold, "LUNAR: Lightweight underlay network adhoc routing", Technical report, University of Basel, Switzerland, January 2002.

[22] Mario Gerla, Xiaoyan Hong, Li Ma, and Guangyu Pei, "Landmark routing protocol (LANMAR) for large scale ad hoc networks", IETF Internet Draft, November 2002 (work in progress).

[23] Kaixin Xu, Xiaoyan Hong, and Mario Gerla, "An ad hoc network with mobile backbones", In Proceedings of the IEEE International Conference on Communications (ICC), New York, NY, April 2002.

[24] Kaixin Xu and Mario Gerla "A heterogeneous routing protocol based on a new stable clustering scheme", In Proceedings of the Military Communications Conference (MILCOM), Anaheim, CA, October 2002.

[25] C. Perkins, E. Belding-Royer and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing", Network Working Group, IETF RFC 3561, July 2003.

[26] Verinet group, Home page: http://www.cis.upenn.edu/verinet.

[27] F. Xue and P. R. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks", Wireless Networks, 2(10):169 – 181, March 2004.

[28] IEEE, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications », IEEE Standard 802.11, 1999.

[29] WIFI-for-UML, home page, http://www.auto.ucl.ac.be/~guffens/uml-wifi/

[30] V. Guffens, G. Bastin and O. Bonaventure, “An Emulation Infrastructure for Multi-hop Wireless Communication Networks”, http://www.auto.ucl.ac.be/~guffens/uml-wifi/download/uml_simulator.pdf, Internal report, 2004.

[31] Mobility Simulation in a Wifi Network of UML Machines, by V. Galtier http://www-lor.int-evry.fr/~galtier/MobUML/

[32] Tutorial User Mode Linux, by V. Galtier: http://www-lor.int-evry.fr/~galtier/TutorialUML/