Address Autoconfiguration in Ad Hoc Networks

by
Bachar Wehbi

Revised by…………………………………………………..……………….…………

Virginie Galtier
Associate Professor

Institut National des Telecommunications -INT-
Departement Logiciels Reseaux -LOR-
Internal Report

May 2005

Table of contents

List of figures

List of tables

I Introduction

Address configuration is an essential phase before network nodes could communicate. For traditional networks, this issue was dealt with by the introduction of Dynamic Host Configuration Protocol (DHCP) and the dynamic configuration of IPv4 Link-Local Addresses.

For Ad Hoc networks, neither approach is not 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.

II Traditional Address Configuration Approaches

II.A 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 there is no DHCP server. The DHCP relay acts as an intermediate between the server and the client to allow DHCP messages to cross routers, thus it should be configured with the IP address of the server.

II.B 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, 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.

III Constraints in MANET Scenarios

In contrast to wired LAN networks, where a broadcast message is able to reach all nodes on the link, the wireless ad hoc 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 MANETs is the energy and bandwidth constraints. Ad hoc 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.

IV Requirements of Address Autoconfiguration

Any address autoconfiguration mechanism should address the following requirements:

- Topology change: ad hoc nodes are mobile and could join and leave the network at any moment without notification. This dynamism of network topology should be considered when designing an autoconfiguration mechanism.

- Network partitioning and merging: during its lifetime, an ad hoc network could be divided in two or more disconnected networks. These partitions or other mobile networks could remerge later. The autoconfiguration protocol should be able to deal with these situations and the resulting address conflicts or address leaks.

V Classification of Address Assignment Algorithms for MANETs

Address assignment in mobile ad hoc networks 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.

V.A 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.

V.A.1 Agent Based Addressing

The Agent Based Addressing proposed in [3], is an autoconfiguration protocol based on a centralized allocation table. It's designed for IPv6 MANETs and supposes the uniqueness of MAC addresses.

In this protocol only one node, the Address Agent (AA) is allowed to assign addresses to requesting nodes, thus it should be always reachable. The AA maintains the allocation table containing already assigned IP addresses with their corresponding MAC addresses and lifetimes.

a. **Protocol operation**

The AA periodically floods the network with “Verify” packets that contain assigned addresses (they do not specify the utility of putting the address list in the Verify packet). When a configured node receives a Verify packet, it responds with a “Confirm” packet to indicate its presence in the network and to allow the AA to refresh the address entry lifetime.

When a new node initializes, it should wait a certain time for a “Verify” packet before requesting an IP address. The address request is sent in unicast from the new node towards the AA. When receiving the request, the AA builds a new 80 bits long IP address based on its MAC address and the requesting node's MAC address. Then it sends the IP address to the requesting node that configures its interface and could then communicate with other nodes in the network.

The protocol specifies a mechanism to dynamically elect the AA so that the network could survive in case of AA departure. Each node waits a specified period of time and expects to receive a “Verify” packet within this time period. If the node does not receive a Verify, it concludes that there is no AA in the network and considers itself as the new AA. This could happen if the node is the only one in the network or if the existing AA has left the network.

To distinguish between different networks, the AA constructs a “Network ID” derived from its MAC address and floods it with the Verify packets. When two networks merge, the AAs will notice the presence of each other by the reception of the Verify flood. The AA with fewer entries in its table should change its state and concerned nodes should request the new AA for IP addresses which leads to unnecessary address changes.

b. **Problems of this protocol**

Even if this protocol guarantees no address duplication, it has many problems.
First, at the initialization phase, the new node is involved in a unicast communication with the AA even that it has no IP address to initiate the communication. How this problem could be alleviated is not mentioned.
Second, it generates a high overhead by the periodic flood of “Verify” messages and their corresponding unicasts from each node toward the AA. In addition, Verify packets contain the complete list of the configured nodes which is not necessary to accomplish the required functionalities.
Third, the protocol is too centralized for a MANET by its high dependency on the AA and does not specify a mechanism for backup. At the same time the address generation is dependent on the MAC address of the AA which leads to unnecessary address changes whenever AA changes in case of network partitions, merges or AA departures.

V.A.2 MANETconf

In contrast with "Agent Based Addressing" where only one node is responsible for assigning addresses and maintaining the allocation table, the idea of 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.

a. Protocol operation

In MANETconf, each node has the possibility to assign new addresses since it holds the allocation table. When a new node wishes to join the network, it broadcasts (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 then chooses a free IP from its allocation table and floods the whole network to have the permission to assign the chosen address. 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, it 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 scratch a certain number of times. If the initiator detects that one or more nodes did not reply, it re-contacts them by unicast reclaiming their permission. Two cases are envisaged. If the concerned node is still connected, it will reply and the initiator could then continue the configuration process. If the concerned node has left the network, the initiator will not receive a reply, thus it concludes after many attempts that the node has left the network and floods this information to inform the whole network about this departure.

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. When a network get partitioned, one partition will conserve its network ID (lowest IP address and identifier) and acts like nothing happened, the other will detect the partitioning with the first IP assignment; only then, it will know the new node with the lowest ID that will generates the new network ID and floods it within the network.

When two or many nodes come within communication range, they exchange their network identities. If the received network identity is different than the nodes network identity, then a network merge is detected. In this case, these networks exchange their different allocation tables.

For example, if node A and node B detects that they belong to different networks, they exchange their allocation tables and A (B) floods the allocation table of B (A). This will allow to all nodes to update their allocation table and to detect locally address duplication. For each duplicate address, one of the two conflicting nodes should release its address. They indicate that it should be the one with fewer TCP connections (they don't specify how to detect the node with fewer TCP connections).

b. Advantages and problems

The advantages of this protocol are that it guarantees address uniqueness and it is totally distributed in term that each node has the possibility to assign new addresses. In addition, it generates no unnecessary address changes when networks merge because only nodes involved in duplication release their IP addresses.

The problems of this protocol are its high complexity in term of communication, table maintenance and synchronization. The mechanism for assigning new addresses is bandwidth consuming; it consists of a network flood and a large number of unicasts. All nodes should give their permission to the initiator to assign a new address, this could generate large delays. Finally this protocol is very sensible to network losses because of its dependency on unicasts communications.

V.A.3 Prophet

The idea behind Prophet [5] is that in place of maintaining an allocation table and working hard to synchronize them 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 intelligence in this protocol is to choose the good generation function. Such a function should fulfill the following properties:

- 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.

a. Protocol operation

When a new node wishes to join the network, it sends a local broadcast to its neighbors. If it receives no reply, it concludes that it’s the only node in the network and configures itself with a random IP address and a random (or default) state value for the predefined generation function. If the new node receives many replies from its neighborhood, it contacts one of its neighbors for requesting an IP address. The requested node uses the generation function 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 numbers. A detailed explanation of how the address generation is performed is provided later.

When a node leaves the network, address reclamation is not needed because the same number will reoccur in the sequence but this reoccurrence is separated by a long period of time. This separation between the reoccurrence of the same address is the guarantee of address uniqueness or more precisely the high probability of address uniqueness.

When a network becomes partitioned and because the existing sequences are different, the newly allocated addresses will still be different among the partitions. In this case, the address generation will remain the same as it was for the original network. Thus their will be no address conflict if the partitions become merged again.

The problem occurs when different networks merge. Because there is no guarantee that the sequences (IP addresses and state values) in the merged networks are different or even they may have different generation functions, address duplicates could exist. Thus a network ID is needed to differentiate between the networks. The network ID is generated by the first node in the network and is passed to new nodes with address allocation. Network merging is detected with the same mechanism as in MANETconf, but both partitions exchange their generation functions and state values instead of the allocation tables. Then possible conflicts will be computed locally and nodes involved in a conflict will be notified to change their addresses. But how these merged networks handle the future allocations in presence of more than one generation functions is not specified. Another simpler approach requires all nodes of one network to give up their addresses and acquire new ones in the second network; the result will be a high number of unnecessary address changes.

For networks of realistic 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' is the IP address of the node generating the new address.
The example shown in figure 1 illustrates the address generation mechanism. Suppose 'A' is the first node in the network, it chooses a random address 'a' and a state value 'e1=0' (in the figure the underlined number represents the state value of the node). 'B' comes and contacts 'A' for an IP address. Then 'A' increments by one its state value (e1=1) and applies the generation function to compute the address of B, the state value corresponding to the generated address is now 'e2=0'; finally 'A' sends the computed IP address (a+3) and the corresponding state value (e2=0) to 'B'… But it remains to be proven that the proposed function can fulfil the requirements. Even in IPv6, a 64-bit interface ID space is exhausted after 64 assignments by the first node (2^{64}), and f(n) may generate duplicate addresses (new assignments by the first node).

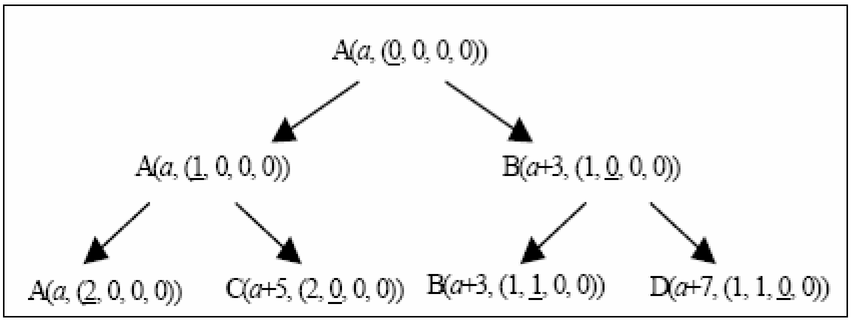

Figure 1 : generation and updates of states of the generation function

Advantages and problems of this protocol

The advantages of this approach are that it generates almost no extra traffic (even in case of network merging the traffic generated is limited). The protocol is very simple to be implemented.

The problems of this approach are:
First, there is no analytic proof that the described function fulfils the necessary conditions. In addition, the approach is only applicable for large address spaces, and the utilization of the available address space is not efficient. Second, the approach does not specify precisely how different networks merge and how they work after been merged. Third, the protocol does not

guarantee the uniqueness of assigned IPs even if the probability of address duplication is very small, and does not specify a mechanism to solve address conflicts in case they occur.

V.A.4 Buddy protocol

In this 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.

a. Protocol description

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.

A new node sends periodically a broadcast message reclaiming an IP address. If the node receives no response, it auto configure with the first IP of the predefined range. If it receives one or more responses, it chooses the first who replies and sends him 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.

If the initiator has no available addresses, it should request its neighbors. 3 possibilities are envisaged:

1. It searches its IP address table for possible one hop neighbor candidates, if it finds no address availability it increment by one the radius of search….
2. It sends a broadcast message to its one hop neighbors, if it receives no reply; it sends a 2 hop broadcast…
3. It searches its IP address table for the node with the biggest block, and contacts it directly.

The synchronization of the address table involves each node to periodically broadcast its address table (this idea is not specified precisely, once it's a local broadcast of the address table, other it's a flood of the current pool). 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 its 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.

b. Advantages and problems of this protocol

Other than the guarantee of address uniqueness, this protocol has the following advantages:

- It generates no unnecessary address changes, only nodes involved in address duplication release their addresses.
- It's totally distributed in term that each node is able assign new addresses.

- The address assignment is only dependent on the involved initiator that is a neighbor of the requesting node, so it’s less sensitive to network losses.
- This protocol is convenient for scenarios with limited address range.

The problems of this protocol are:

- It is complex to be implemented
- The synchronization mechanism is complex, and need high convergence delays.
- The address distribution is not totally “even”, it depends on network concentration.
- The synchronization overhead is high, it requires network flooding.

V.B 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.

V.B.1 Strong Duplicate Address Detection (SDAD)

The SDAD presented in [7] 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.

a. Protocol operation

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.

Problems and limitations of this approach

Even if this approach is the simplest we could imagine it has many problems and limitations. The duplicate address detection performed is 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 is not able to solve this duplication which disturbs the performance of the network. At the same time network merging or simply temporal disconnections are not considered.

This protocol does not guarantee address uniqueness along the network, and duplication probability increases with network size in case of a limited address space.
And finally, it generates high overhead with each node join constituted of several network floods.

V.B.2 Weak Duplicate Address Detection (WDAD)

The WDAD proposed in [8] aim at extending the duplicate address detection mechanism for the whole lifetime of the network. The idea behind WDAD 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 to make routing capable of differentiating between potential duplicate IPs.

a. Protocol operation

Each node generates a key at initialization phase, and distributes it with its IP address in all routing messages. This key will be used to detect duplicate IP addresses.
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).

b. Problems and limitations

The main drawback of WDAD is its dependency on the routing protocol. It requires some changes to the routing layer to support the introduction of the key identity. Each node will be identified at the routing layer by a kind of virtual address consisting in the combination of the IP address and the key value. In addition, WDAD detects address duplication based on local routing information, thus it is totally adapted to proactive routing where each node maintains a complete routing table. For reactive routing, it is not the case; the nodes cache partial routing information for only ongoing and relayed connections which reduces the possibility of detecting in moderate delays address duplication.
For the overhead, WDAD requires no additional traffic for the autoconfiguration mechanism, but the price is traffic overhead caused by the integration of the key value in routing packets.

V.B.3 Passive Duplicate Address Detection (PDAD)

PDAD [9] is a duplicate detection mechanism designed for link state routing protocols. The idea behind PDAD protocol 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.

a. Protocol operation

With proactive routing, the nodes periodically flood the network to inform other nodes about their neighbourhood. These control packets contain sequence numbers to distinguish between fresh and old packets. Based on these information, PDAD analyzes incoming routing packets to detect address duplicate. In [9] a complete list of mechanisms used to passively detect address duplication is presented, next I will explain one of these mechanisms "Passive Duplicate Address Detection Based on Sequence Numbers" just as an 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.

b. Advantages and limitations

The advantage of this protocol is that no additional overhead is generated; but it requires complex analysis of the routing information and is applicable only for proactive routing protocols.

V.B.4 Ad Hoc IP Address Autoconfiguration

The Internet draft presented in [10] 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.

Like in WDAD, 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 a new IP address.

This draft could be considered as the most mature proposal for stateless address autoconfiguration.

V.C 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.

V.C.1 Hybrid Centralized Query-based Autoconfiguration (HCQA)

The HCQA protocol [11] is the first hybrid approach proposed. It utilizes SDAD mechanism along with a centrally maintained allocation table in order to improve address consistency.

a. Protocol operation

At initialization phase, a node chooses two addresses, a temporary and a tentative one, and performs SDAD exactly as explained in V.B.1. 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 from the AA, 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.

When nodes initialize, the first node that obtains a unique IP address becomes the Address Authority (AA) in the network. The AA chooses a unique identifier (ex: its MAC address), and broadcasts it periodically to identify the network. If a node does not hear any AA advertisement for a certain period, it considers that there is network partitioning and becomes the new AA and generates a new network identifier. When a node hears a new network ID, it must register its address with the new AA, thus no address change is needed. Network merge is detected by the presence of two network IDs. In this case, only the AAs are involved in detecting address conflicts by exchanging their different tables.

To reduce the centralization at the address authority, the protocol specifies a mechanism to backup the address authority's address table. To do so, the AA picks the first node that has registered its address as the "Address Authority Backup". Every time a new node registers its IP address with the AA, the AA sends an update with the new information to the address authority backup.

b. Advantages and problems of this protocol

This protocol adds robustness to the SDAD mechanism, by guaranteeing duplication detection. At the same time it proposes an effective mechanism for detecting and handling network partitioning and merging.

On the other hand, this protocol has some problems. First, the overhead generated by duplication detection and the periodic floods of the AA is very high. Second, the address autoconfiguration is dependent on a central entity which requires all nodes to register by unicast their addresses. This mechanism increases the autoconfiguration delay and the sensibility on network losses.

VI Autoconfiguration and EcoMesh

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.

VI.A The EcoMesh model

We are placed in a scenario where an ISP plans to extend its meshed hot zone's coverage area by a collaborative ad hoc extension reserved for its own clients. Distant clients will be able to reach the Meshed side by using bandwidth resources offered by intermediate clients acting as relays. Thus the collaborative side is very important to make the network survivable. Our study will be limited to the ad hoc extension. Given that the offered quality of service degrades with each additional hop, we limit the number of ad hoc hops to maximum 4 or 5.

In such a scenario, we can assume that the probability to meet the same users is very low, thus the intra-ad hoc communications are negligible compared to the communications with the external network (Internet, services provided by the ISP???). This must not eliminate the possibility of communications between ad hoc users; and our solution has to take in consideration the possibility of such a communications.

Accordingly, the network will have the meshed backbone as a stable part that will be always reachable (if not the ad hoc extension will lose its reason to be). This stable backbone will serve us in planning for security and autoconfiguration.

VI.B Requirements for EcoMesh's Address Autoconfiguration

When placed in the EcoMesh context, address autoconfiguration should fulfil 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. To do this, meshed routers should have a global vision of the ad hoc extension.

- Address limitations: in the EcoMesh context we can imagine two scenarios concerning the available address space. First, a limited range of real addresses dedicated for ad hoc users; hence it must be carefully distributed and address leaks have to be detected and treated in a reasonable time. Second, a large range of private addresses, in this case address translation is needed to globally connect the network. Deciding whether to use public limited range or private large range influences planning for autoconfiguration.

- 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.

VI.C 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 identify the major research actors in this field and to note the state of advancement of this work (see table 1). It should be mentioned that none article provides formal specification of the protocol, only informal description is presented.

	Univ. or Lab/ Date of Publication	*Implementation*	*Simulation*	*Modification at MAC layer*	*Approach*
Agent Based Addressing [3]	Aachen University Sept 2002	None	NS	Unspecified	Stateful
MANETconf. [4]	Univ. of Dallas INFOCOM 2002	None	NS	Yes	Stateful
Prophet [5]	Michigan State Univ. Hong Kong Univ.	None	NS	Yes	Stateful
Buddy Protocol [6]	Univ. of Texas MILCOM 2002	None	NS	Yes	Stateful
SDAD [7]	Nokia Research Univ. Santa Barbara IETF draft, Nov. 2001	None	None	No	Stateless
WDAD[8]	Univ. of Illionis MobiHoc 2002	None	NS	No	Stateless
PDAD [9]	Univ. of Karlsruhe IEEE WCNC 2003	None	NS	No	Stateless
Ad hoc IP @ Autoconf. [10]	Univ. of Minnesota IETF draft, February 2005	Work in Progress	None	No	Stateless
HCQA [11]	Univ. Santa Barbara June 2003	None	NS	No	Hybrid

Table 1: Underlying 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 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 incitative mechanisms. The overhead should be described by the required bytes per node; but 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 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 diameter of the network
- l: the average number of neighbours
- T: the period of synchronisation, 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
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 2: Performance comparison between existing protocols

For example, if we take the Agent Based Addressing; it requires a request/reply communication with the Address Authority for each address assignment. Of 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.
Last, we compare the existing approaches based on evenness, routing dependency, distributed operation, address uniqueness and stability (see table 3).

- Address Evenness: this is an important metric in the case of EcoMesh if we consider that the available address space is limited so this metric gives an indication of the effectiveness of the address distribution. An even distribution means 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, 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 may tolerate a certain level of centralization but at the same time we should consider the potential effects of such centralization. For example, if the address assignment will be centralized, the network overhead will be higher and the configuration delay too; this would be problematic in a mobile environment.

- Address uniqueness: address duplicates may occur if two networks merge or in the address assignment phase with stateless approaches. In the EcoMesh context address duplicates are not acceptable because other than the network perturbation, it may have a negative effect on the security or the incitation 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 users' 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
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	Integrated within the routing protocol	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 : characteristic comparison between existing 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 4 or 5 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.

VII Conclusion

In this report we have presented the existing address autoconfiguration protocols, and classified them according to the way they maintain the available addresses. Then we have defined the EcoMesh model (the definition is flexible and may be changed) and the requirements for autoconfiguration in the EcoMesh context. And finally we have conducted a comparison between the available approaches and tried to approach this comparison to our context.

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] 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.

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

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

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

[11] 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.