Evaluating Automatic Pools Distribution Techniques for Self-Configured Networks

Reinaldo Gomes
Systems and Computing Department
Federal University of Campina Grande, Brazil
reinaldo@dsc.ufcg.edu.br

Ricardo de O. Schmidt
Design and Analysis of Communication Systems
University of Twente, The Netherlands
r.schmidt@utwente.nl

***Abstract*—Next Generation of Networks (NGN) is one of the most important research topics of the last decade. Current Internet is not capable of supporting new users and operators' demands and a new structure will be necessary to them. In this context many solutions might be necessary: from architectural definitions to new protocols. Addressing protocols are a specific example of protocols which should be defined to support NGN requirements. One special required characteristic is automation of addresses assignment to facilitate networks operation and design. Many addressing levels can be considered, however, proposed solutions are usually restricted to local networks addresses distribution. In this paper we present an analysis over automatic address distribution to networks, allowing a correct local addresses' assignment. Two allocation techniques are presented and evaluated to present the benefits of this kind of mechanisms. Finally, conclusions about the proposed methodologies and the protocols applicability are discussed.**

I. Introduction

In future networking, a whole new level of dynamicity and heterogeneity is expected [1]. Following this new networking paradigm, future networks' protocols must enable cooperation of heterogeneous networks that might belong to different operators or use incompatible technologies [2]. This cooperation should be transparent, established on demand, and thus support "plug-and-play" operation, i.e., no previous configuration should be required when forming the networks.

New communication scenarios are expected to be deployed [3] in order to use new characteristics that might be supported by these new protocols. In this context, situations where new networks are spontaneously created according to users' needs will happen, creating a new networks' establishment and connection model. Networks' connectivity is supposed to be unpredictable and, consequently, agreements defining how the domains might exchange information among them should be dynamically created to represent topology and connectivity modification of the environment.

An example of these possible scenarios is a Personal Area Network (PAN) which is created by the devices carried by a user (i.e., laptop, cell phone, PDA, etc.) [4]. In a certain moment the user needs to access some resource which is not possible to be offered by local devices, needing a valid set of addresses to correctly access the other networks. To solve this question, a mechanism which could adapt addresses distribution to these dynamic demands would be extremely important to support automatic networks spontaneous creation.

The rest of this paper is organized as follow. Sections II and III introduce address pools distribution and issues of their dynamic operation. The proposed and evaluated strategies are described in Section IV. The results obtained through our experiments are presented and analyzed in Section V, pointing out main strenghts and drawbacks of evaluated mechanisms. Finally, in Section VI, conclusions about the presented work are drawn.

II. Address Pools Distribution

Normally, networks might configure their elements by assigning addresses to hosts (nodes). It requires a set of addresses to be distributed among the network elements [5]. This set of addresses is known as a pool of addresses. A pool is a range between two addresses. It only contains addresses which are valid and unique. By using a control policy, the pool allocation can be autonomous [6]. But, policies might require manual intervention for initial configuration. In addition, simple automatic pools allocation will not necessarily optimally distribute pools improving network performance.

Allocation mechanisms aim to configure subnets with adjacent addresses or pools. It means that they try to avoid allocation of discontinuous pools to the same subnet [7]. For instance, consider multiple starting networks (or subnets), all of them requesting pools according to their respective needs. When no allocation strategy is implemented, there is a high chance that these networks are configured with disjoint pools. As a consequence, it would demand from intermediate devices the separation of entries for each subnet in routing tables. Generally, it is less efficient than having a single entry in the routing table, pointing to all hosts connected to the same subnet. It would only be possible with allocation of adjacent pools to a same subnet, allowing a single representation in the routing table.

Adjacent pool allocation can be performed by several different strategies. The implemented strategy will impact in both the way allocation is done and the influences it causes in different network layers [8]. In the routing level the impacts are in the construction and maintenance of routing tables. Therefore, an inefficient allocation can severaly impact in the routing performance by, for example, increasing the amount of information in the routing table.

To deal with this problem and to allow a completely dynamic distribution of pools, two approaches are proposed in this work. An example of the operation of an automatic address pools allocation technique is when a new network starts to operate and needs a valid pool to configure the devices that are composing it. Using the current technologies it would be necessary configuring all the servers with their specific pools of addresses manually and when the network need more addresses the administrator must obtain more addresses and configure them [7]. On the other hand, with an automatic allocation networks can request address pools dynamically according to their demands, adapting to the particular needs that they may find during their operation.

III. Mechanism Overview

The proposed mechanism defines two allocation techniques that might be used in order to distribute addresses pools in a highly dynamic environment, allowing the networks to obtain the pools necessary to address all their clients and at the same time aggregate as much as possible the allocation information and, consequently, the routing information.

These mechanisms are planned to improve the pools assignment in a global scale organizing the allocated addresses in a way that they could be distributed following some disciplines and also having the ability of considering policies to control the addresses distribution and adapting to the particularities of each network.

A typical situation to apply the proposed mechanism is the automatic configuration of a set of networks that are connecting with each other and need to obtain a pool of valid addresses to configure their local network elements.

Initially a network or a set of networks is supposed to exist and will be configured with the initial addresses pools. This configuration might not be performed by a negotiation, needing the definition of policies to allow the configuration of these servers without any human intervention. These initial servers can be compared to DNS root servers which present less dynamism and more computational capabilities, and because of it are expected to be responsible by this initial distribution of the addresses using some pre-defined policies.

Having these initial pools distributed among the root servers, new networks that are starting to operate should receive a pool of addresses from an already connected network to be managed by the addressing server of this new network. This distribution is performed in a hierarchical way, but there are no guarantees that no more addresses will be necessary to the networks, needing an allocation mechanism that improve and facilitate this allocation.

This distribution of the addresses can be performed following many different strategies and according to the selected algorithm the results might be different to the allocation itself and to the influences that this addressing mechanism will have in different levels. The most important of these influences is in the routing level, since the allocation will influence in the construction and, consequently, in the structure of the routing tables, so a bad structured allocation can badly influence the routing mechanism by the increase in the amount of information stored in the routing tables.

To deal with this problem and allow a completely dynamic distribution of pools two approaches are proposed in this work. Following these approaches the networks are able of allocating their pools to new networks in a more structured way and then avoid problems in routing as described in the previous section.

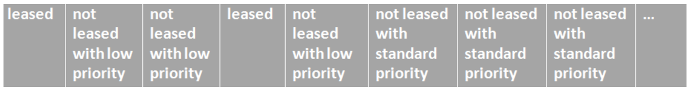
An example of the operation of this mechanism is when a new network starts to operate and needs a valid pool of addresses to configure the network elements that are composing it. Using the current technologies it would be necessary configuring all the servers with their specific pools of addresses and when the network need more addresses the administrator must obtain more addresses and configure them. On the other hand new nodes just cannot be added using valid IP addresses. Using the proposed mechanism the networks are able of requesting pools dynamically according to their demands, adapting to the particular needs that each network can find during their operation.

When a new network (a spontaneous network created according to users' demands) starts to operate it must request a pool of addresses to an already connected network that have available pools. This network will them allocate one or more pools to the new one and make a pre-reservation of other pools that might have the preference of being allocated by this network instead of other networks, grouping the addresses allocated to all the network and avoiding the influence which discontinuous addressing will have in communication.

IV. Pools Allocation Mechanism

Considering the questions already exposed, it's easy to note that only allocating the pools is not enough to allow the network to operate in an optimized way. Allocating pools without any special control can solve the dynamic allocation problem but it brings us another problem: routing table would increase, since to every new pool a new entry would be necessary to reach the nodes allocated within the new range of addresses.

Focusing on controlling the pools allocation, two techniques were defined to improve pools assignment in a global scale, organizing addresses distribution by following



leased	not leased with low priority	not leased with low priority	leased	not leased with low priority	not leased with standard priority	not leased with standard priority	not leased with standard priority	...

Fig. 1. Example of Consecutive pools distributions with reservation

some techniques, and also having the ability of considering policies to control the addresses distribution and adapting to the particularities of each network. A second objective of these techniques is to avoid routing tables' size problem.

These approaches are based on two basic structures that are intended to support a diversity of situations, from stable networks, that can rigidly control their needs of addresses, to high dynamic networks, were this control is not possible. In the first situation a consecutive allocation might suffice, but in the second case an allocation where it is possible to spread the addresses, might adapt in a better way to the dynamic aspects of the networks.

A. Consecutive Allocation with Neighbor Pool Reservation

The first approach is based on a reservation of neighboring pools and can be applied with both: fixed or variable pools size. In this initial approach, at the moment of the pool allocation, the reservation of a set of pools, that are consecutive to the one that will be allocated, might indicate that the "owner" of the allocated pool have the preference to use the reserved ones. The amount of addresses that will be reserved to each allocated pool might be different and could be defined using a set of policies that will evaluate the stability of the network and its possible increase. This reservation could be done, for example, by modifying the priority of the sequential pools (that were not leased) of the allocated one, to a lower priority in being allocated.

Fig. 1 presents an example of this mechanism. Suppose that the first pool was allocated for a new network. The defined policies verify that, considering its characteristics, the reservation of two pools may be enough to supply future assignments in this new network. Based on that, the two subsequent pools, not leased yet, are marked with a lower priority to be allocated for a different device, indicating the preference to be allocated to extend the range of the new network. The number of sequential allocated pools when a request is received can be determined by monitoring of the environment, which allows the prediction of the demand of each subnet.

After this first allocation, a second network might also request a pool. To this second network the same verification is performed in order to define the size of the resources reservation. After allocating the pool and reserving the very next one, the approach will continue its operation by receiving new requests and allocating the addressing resources until they are available. Only when no more available pools with standard priority exist, the reserved pools will be considered to new requests.

This mechanism facilitates adjacent pools to an already allocated, being allocated when the same network requests more resources. This control intends to avoid the increase in the routing tables, as described as one of the major problems of non-controlled allocation. The main advantage of this mechanism is that not leased pools and the ones with the standard priority are organized in a sequential order, which causes the routing tables to not increase.

B. Spread Allocation with binary-based division

In the second proposal, the pool is divided in two different parts every time an allocation should be done. By dividing an existing pool in two parts, each new network would initially have half of the available addresses reserved to it. To perform this allocation, the approach attempts to allocate pools to new networks as far as possible from the previous allocated ones in the addressing range, which avoids a bad calculation of the necessary resources amount and consequently allocation non-consecutive pools to the same network. The following description explains the general mechanism and how it avoids the increase of routing tables.

Initially, it is necessary to divide the original pool that will be allocated in a set of sub-pools, preferably with the same size. Considering that a server, which will be responsible for allocating addresses, has a pool P1 with size S1, this pool will be divided in n sub-pools with the same size S1/n (or similar sub-pools sizes). The mechanism could also be applied for the allocation of pools with variable size. But with a fixed size, the management of the pools is easier. In order to facilitate the understanding, the approach's description will be based on allocation of fixed size sub-pools.

When the first request for pool is received by the server, it allocates the very first sub-pool in the range and considers that all addresses that follow the allocated one are pre-reserved to the same network for future allocations (Fig. 2(a)).

If new pools are requested by the first network, it should use the pools pre-reserved to the allocated one (X1). As all pools are pre-reserved to X1 the new pools will be allocated sequentially to the right until it use all available addresses. When a second request for pool is received, from a different requester, the space reserved to X1 is be divided in two equal parts and it allocates the farthest sub-pool from X1, to the new network X2 (Fig. 2(b)). This way, the allocation for the network X2 will grow sequentially towards X1 reserved space (i.e., to the left).

The main idea of this approach is to make the sub-pools allocation, for different networks, grows towards each one. Therefore, it makes easy the sub-pools aggregation when allocating them to the same server, and also will allow a "free-growing" of the allocated space to a determined network, avoiding the need for the verification of the network's characteristics in order to calculate the amount of addresses to be reserved. Every time that a request for

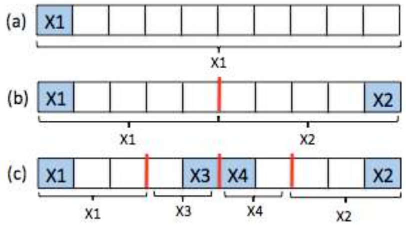


Fig. 2. Example of allocation sequence using spread allocation with binary division technique

pool is received at the server responsible for managing the resources, the part of the pool that is already reserved to a specific network is divided in two equal parts and each network will have half of the available addresses pre-reserved to them.

Continuing on the example of Fig. 2, when a third request is received from a new network, the area reserved to X1 is divided in two equal parts and the farthest sub-pool from X1, considering only the divided area, is allocate to X3 (Fig. 2(c)). This way, the allocation to the third network will grow towards X1. Now, both the allocation to X2 and to X3 grows to the same direction (i.e., to the left). Therefore, when receiving the fourth pool request from a network X4, the area reserved to X2 will be divided in two equal parts and the farthest sub-pool from X2 is allocated to X4. This way, the allocation to X4 will grow towards X2 (Fig. 2(c)).

While receiving pool request messages from new networks, the server keeps the same methodology for the allocation. That is, the server chooses an area that is already reserved to a network (considering pre-defined policies such as the largest reserved area is the one to be split), and divides it in two equal parts. Next, the server allocates the farthest sub-pool from the existing network allocation, so that the new allocation will grow towards the allocation of the already existing network (i.e., the one that had its pre-reserved area divided).

In order to add flexibility to this mechanism, it is possible that the area limiter (red line in the Fig. 2 and 3) is moveable. It permits a network allocates more sub-pools than its initially reserved area. However, considering the structure illustrated in the Fig. 3(a), network X1 can only allocate more sub-pools if network X3 is not using all the sub-pools pre-reserved to it, i.e. those that are part of network X3's pre-reserved area.

Following the same methodology, sub-pools' allocated to network X1 grows towards the sub-pools' allocated to network X3. In the situation that network X1 is already using all pre-reserved sub-pools to it (Fig. 3(b)), i.e. only those that belong to its pre-reserved area, the flexible area limiter allows sub-pools from X3's reserved area be allocated to X1.

Upon receiving a request for pool from network X1, the server checks the pre-reserved sub-pools from child X3's area. If X3 is not using all its pre-reserved sub-pools, it is possible to reallocate one of them to X1, as presented in Fig. 3(c). This way, X3 lends the farthest sub-pool from itself to network X1, which would be the last pre-reserved to be used by X3 if necessary. This methodology guarantees the integrity in the sub-pools aggregation when keeping the sub-pools sequentially allocated. If the network X1 gives the sub-pool back to its higher server, the area between X1 and X3 is recalculated and the limiter returns to the previous position (Fig. 3(d)).

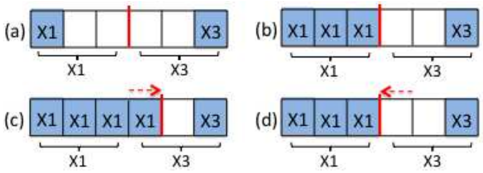


Fig. 3. Example of *X1* extra pools allocation

During this process, in many situations the sub-pool that should be returned is not necessarily the farthest one. The available addresses can be located in an intermediate sub-pool, for example in the second one. However, if network X1 gives the second sub-pool back to the server, it will create a fragmentation in the allocation sequence. To avoid this situation, only the last sub-pools can be returned to the servers (following a Last-Input-First-Output scheme), and all networks/nodes that are currently configured with addresses from these sub-pools should be reconfigured and a reassignment using intermediate sub-pool's addresses should be done, restoring the initial structure and keeping the available sub-pools always together.

V. Experiments

As no address's pools distributions techniques are used in Internet, i.e. networks' addresses are statically allocated in an offline procedure, no other solutions were implemented to be compared with proposed allocation techniques. To have a base value for comparison with proposed allocation techniques a consecutive allocation with no pools distributions strategy was also considered since it would be the worst case of dynamic allocation in routing table's perspective. To all evaluated mechanisms, requesting networks identification was selected randomly, until the server (i.e., allocator) ran out of pools.

Evaluations considered a different amount of available pools. The number of requesting networks is also varied in order to verify how the distribution mechanisms would deal with situations where a higher number of networks are trying to obtain pools and, consequently, the possibility of having consecutive pools allocated to different networks is higher.

A. *Pools Allocation*

In this experiment, the numbers of available pools and requesting networks were varied. In Fig. 4, the results

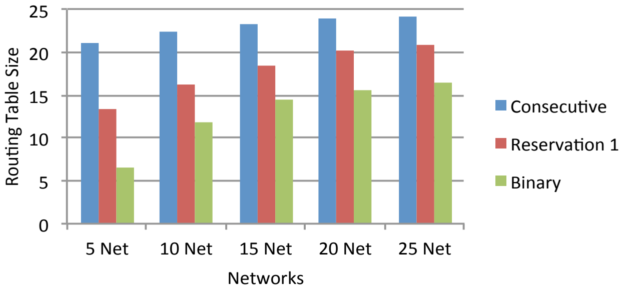


Fig. 4. Routing tables' size with 25 available pools and requesting networks varying from 5 to 25

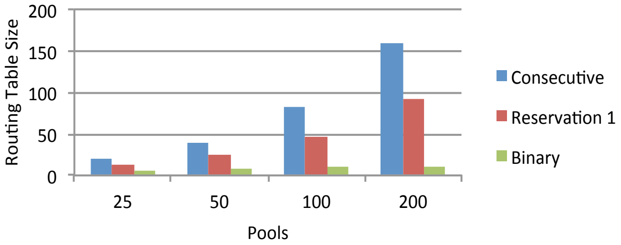


Fig. 5. Routing tables' size for 5 requesting networks and available pools varying from 25 to 200

regarding the average routing tables' size are presented.

Fig. 4 presents the results obtained by the pools allocation mechanism when having 25 pools and changing the number of requesting networks. It's possible to note that when working with 5 networks requesting 25 pools, the consecutive allocation with no technique has created routing tables with more than 20 entries. As the number of requesting networks increase the routing table's size also increases, being very close to 25 entries (which is the maximum number of possible entries when 25 networks request pools). As the requesting network identification is selected randomly, in some situations not all 25 networks will request a pool before the resources exhaustion. This is the reason that the average routing table size is smaller than 25 entries.

Consecutive allocation with Neighbor Pool Reservation of one pool has also increased routing tables' size with more requesting networks. However, this increase is lower than with the allocation with no strategy. With 5 requesting networks routing table had about to 13 entries, instead of 20 with no allocation strategy. Only in with 25 requesting networks it reached the same level of the initial value with no allocation strategy, indicating that this allocation mechanism can reduce the routing table size even when reserving only one pool for each new allocation.

Spread Allocation with reservation based on binary division reached the best results. For 5 requesting networks it had about to 6 entries. In ideal situations it would just create 5 entries, one to each network. But, as requests are not equally distributed to all requesting networks, some of them will need to create more than five entries to store all the information. In this allocation technique the number of entries in the routing tables also increases with more networks requesting pools, but the amount of information that might be stored is significantly lower than with the other allocation techniques.

Beyond the understanding of the general behavior of routing table's size, to the different amount of pools to be allocated, it would also be necessary evaluate how it behaves when considering only the variation on the number of available pools, keeping the amount of requesting networks. This evaluation might indicate how each mechanism can adapt to networks with more resources in order to optimize address allocation and routing operation.

In this case, Fig. 5 presents the results of the evaluation for 5 requesting networks when available pools increase from 25 to 200 pools. In consecutive allocation with no strategy is possible to see that, independently of the number of pools, it continues creating highly fragmented routing tables, since only a small part of the allocated pools were grouped. Allocation with one reserved pool to each new allocation also continues increasing with more pools, but in a lower proportion than presented in allocation with no strategy.

The allocation mechanism that presented the best performance in this scenario was the reservation based on binary division. As presented in Fig. 5, this allocation kept almost the same routing table size, independent of the amount of available pools, having a very discrete increase in the necessary information to routing level.

B. *Routing Table Aggregation*

Only the number of entries existent in the routing table can make a good representation of the resources usage and how the mechanism can adapt or optimize its operation, according to the amount of available resources. To have a representation of the amount of grouped entries in a routing table, the routing table's spreading indicator was defined. It is responsible by calculating the amount of entries that are not aggregated in the routing table.

In Fig. 6, it is presented the aggregation level in the routing table, reached by each of the evaluated strategies for pools allocation. The consecutive allocation basically kept the same aggregation to the amount of requesting networks, independent of the number of available pools, i.e. in Fig. 6 it only had about to 20% of the entries aggregated, independent of the amount of available pools.

The reservation strategy had a smooth increase in the number of aggregated entries in the routing table, having nearly to 50% of the routes aggregated to the experiments executed with 5 requesting networks (Fig. 6). In the experiments with more requesting networks the increase of the aggregation level was higher. However it does not mean that the mechanism had a better adaptation to these situations. Actually, such increase indicates a worse per-

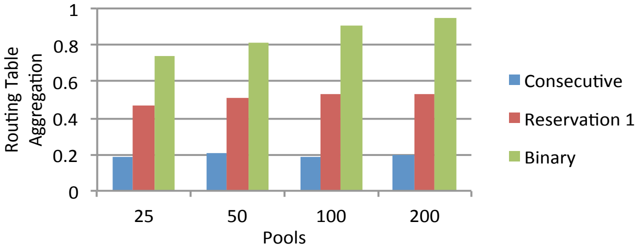


Fig. 6. Routing tables' aggregation to 5 networks and available pools varying from 25 to 200

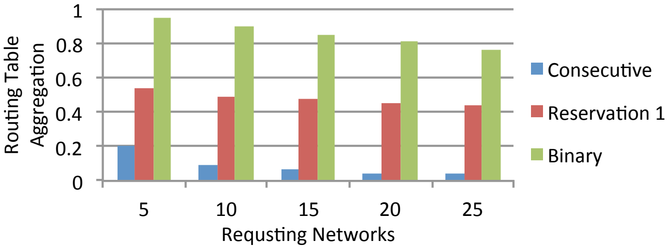


Fig. 7. Routing tables' aggregation to 200 available pools and requesting networks varying from 5 to 25

formance in experiments with more requesting networks, since the routes aggregation starts in a lower level, and only with 200 pools it is closer to the same results obtained with 5 requesting networks.

Allocation by reservation based on binary division presents a considerably increase in all the evaluations. It happens because as the number of available pools increases, the number of pools which will be reserved to each network is bigger and the probability of allocating subsequent addresses to the same network is higher.

In general, the results indicate that with a higher number of requesting networks the aggregation level of the routing tables is reduced. In the case of the binary division mechanism this reduction occurs because the more networks requesting pools, the reduced the number of pools that can be reserved to each network . As consequence there is a higher probability of exhausting the reserved pools (pre-reserved area) and then it is needed to allocate non-sequential pools, fragmenting the entries in the routing table. With the consecutive allocation with no strategy and one pool reservation mechanism, this increase happens because the more networks request pools, the more difficult it is to have not allocated and consecutive pools to be assigned, given that there is no reservation of pools or only one pool is reserved in each allocation.

A second situation that can be studied is how routing tables' aggregation behaves in function of demanding networks, as presented in Fig. 7. In this situation is possible to see that all allocation strategies had a lower aggregation level with more networks requesting pools. This is an expected result, since, the greater the amount of networks, the higher the probability of having networks which exhaust their reserved pools and needed to allocate a new range of addresses.

Consecutive allocation kept the worst results of the evaluated techniques, reaching values very close to zero with 25 requesting networks. Binary division still presents the best results, with the reservation technique having intermediate values, but the reservation had a lower decrease of the aggregation level than binary technique. This indicates that reservation technique might have a better scalability to this particular situation; however, it doesn't compensate the poorer results compared to binary division technique.

VI. Conclusion

This paper described some issues related with networks autonomous addressing, which comes to support autonomous networks in general. Two mechanisms responsible by distributing addresses pools among networks were evaluated. The experiments compared the mechanisms verifying the gains they could proporcionate mainly when networks operate unpredictably, i.e. requiring the support of an autonomous addressing management. However, address resources distribution alone does not guarantee that network operation will be improved.

The evaluation scenarios presented a substantial variety of operations considering routing tables size and aggregation. The evaluated strategies for pools allocation achieved significant and promising results by, for example, substantially reducing the size of routing tables and improving routes aggregation at a fairly representative rate when compared unplanned allocations. As future work we aim the definition of a standardized protocol for address pools allocation and the joint evaluation of addressing protocol with pools management.

References

[1] N. Niebert, *Ambient Networks: a framework for mobile network cooperation*, in proc. of the 1st ACM Workshop on Dynamic Interconnection of Networks (DIN), pp. 2–6, 2005.

[2] 4WARD, *The EU-FP7 4WARD Project (#216041)*. Available at: <http://www.4ward-project.eu>. Accessed in: April 2011.

[3] K. Weniger and M. Zitterbart, *Address autoconfiguration in mobile ad hoc networks: current approaches and future directions*, IEEE Network Magazine, Special Issue on Ad Hoc Networking Data Communications & Topology Control, vol. 18, issue 4, pp. 6–11, 2004.

[4] E. Baccelli, *Address Autoconfiguration for MANET: Terminology and Problem Statement*, IETF Internet Draft, 2008.

[5] K. Manousakis, *Network and Domain Autoconfiguration: A Unified Framework for Large Mobile Ad Hoc Networks*, Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of Maryland, 2005.

[6] D. C. Verma, *Simplifying Network Administration using Policy-Based Management*, IEEE Network, vol. 16, issue 2, pp. 20–26, 2002.

[7] J. Hawkinson and T. Bates, *Guidelines for creation, selection, and registration of an Autonomous System (AS)*, RFC 1930, 1996.

[8] T. K. Forde, L. E. Doyle and D. O'Mahony, *Self-Stabilizing Network-Layer Auto-Configuration for Mobile Ad Hoc Network Nodes*, in proc. of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), vol. 3, pp. 178–185, 2005.