A Cooperative HIde and SeeK DiscoverY over In Network Management

Lucas Guardalben, Susana Sargento and Paulo Salvador
Universidade de Aveiro/Instituto de Telecomunicações
Aveiro, Portugal
guardalben,susana,salvador@ua.pt

Vitor Mirones
Portugal Telecom Inovação S/A
Aveiro, Portugal
mirones@ptinovacao.pt

***Abstract*—Future networks are expected to become more complex and dynamic than today's networks. In these scenarios, a centralized approach for network management is extremely difficult and contains serious scalability problems. An alternative to solve these problems is to introduce the concept of In Network Management (INM), where each INM entity, a network node, has the autonomy to self-govern its behavior. In this concept, each INM entity participates in a distributed management process, which requires cooperation between nodes to monitor, analyze, decide and act upon the network. However, to ensure the communication of nodes in In Network Management process, several phases need to be performed, such as, discovery and bootstrapping, exchange of network information, dissemination of local decisions and final decision dissemination for enforcement. In this paper we address the discovery and bootstrapping, and exchange of information between nodes, towards the complete INM process. We propose *Hide and Seek* (H&S), a new algorithm for network discovery, and information propagation and synchronization. The results show that H&S spends fewer cycles to discover all nodes and records lower messages overhead for information synchronization when compared to non-controlled and probabilistic (gossip) flooding dissemination.**

I. Introduction

In the future, different networks and technologies will interact in a global network of networks. Multiple access technologies will be simultaneously available. The choice of access network could depend on technical (achievable bitrate, maximum delay) or non-technical (cost) decision criterions.

Network elements and terminals will vary from the simplest, e.g. a sensor, to the most complex, e.g. a server or a mobile router. The role and capabilities of such devices in the management overlay will depend on their characteristics. Cooperation and delegation between nodes will be required. Changes in the network can occur naturally when a node arrives or departures from the network, or can be caused by a malfunction somewhere in the network. There are similarities in the handling of both processes. The ability to adapt to changes is nowadays performed by a traditional Internet management, through centralized approaches. However, they should be provided without the need for human intervention.

The traditional Internet management approaches use external management that resides outside the network on servers and stations that interact using standard protocols, such as SNMP (*Simple Network Management Protocol*) or CLI (*Command Line Interfaces*). As depicted in Fig. 1 , the traditional management generally uses dedicated servers to take decisions and actions.

The vision of *Autonomic Network Management* [1] is being pursued by researchers as an answer to the previous problems. This is the capability of network entities to self-govern their behavior within the constraints of business goals that the network, as a whole, seeks to achieve. To address all these concepts, new Internet architectures need to be designed, following the GENI initiatives in the US of Clean Slate design approaches to re-build the Internet. In this sense, a new paradigm to ensure autonomic network behavior is In Network Management (INM)[2], [3], [4] that is based on a distributed control approach. This paradigm was studied in the scope of 4WARD project [5], [6] and describes initiatives that facilitate the embedding of distributed management functionalities over INM entities. As opposed to traditional management, in INM each entity interacts with its peers with the ability to take decisions based on the knowledge from the other elements [2] (Fig. 1).

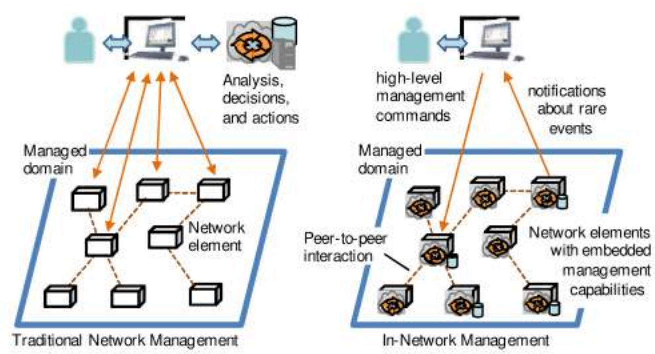

Fig. 1. Traditional (left side) versus In Network Management (right side) [4]

Consequently, for the nodes to know each other and to communicate information and management decisions in an efficient way, some phases are required to provide the means for the nodes to cooperate. Some of the phases comprise discovery of nodes and bootstrapping of the network, exchange of network information between nodes, dissemination of local decisions from the nodes to other nodes, and final decision enforcement dissemination to perform distributed enforcement of the cooperative decisions (section II). After completing all these phases, it is possible, in terms of communication, to embed INM concept over the network through INM architectural entities. For the discovery, bootstrapping and exchange of network information, we propose the Hide and Seek (H&S) approach, a new algorithm for network discovery, bootstrapping, and information propagation and synchronization. H&S is essential to ensure the communication to propagate the relevant and sufficient information between INM entities, necessary for the decision processes.

The rest of this paper is structured as follows. Section II summarises the In Network Management concept. Subsequently, the related work is described in Section III. Section IV presents the *hide and seek* algorithm. Next, the simulation setup is depicted in Section V and results are discussed in Section VI. Finally, Section VII summarises the important conclusions and future work.

II. In Network Management (INM)

The INM paradigm is based on a distributed control approach. This approach provides continuous interactivity between nodes in order to exchange information about each node and its traffic (and therefore the network). This information will allow the network to make automatic decisions, reacting to network changes (such as link failures, load variations) and continuously optimizing the network resources (in both physical and virtual networks, to users and services) according to optimization mechanisms. Furthermore, to ensure nodes knowledge and communication in INM process, four phases are envisaged, such as the discovery and bootstrapping, exchange of information, dissemination of local management decisions and final management decision dissemination for enforcement.

In the first phase, the INM entities need to communicate to discover the network nodes and information, their roles, and how they should operate. As will be referred in Section III, this communication process should have low cost in a distributed way. In this sense, we consider the bootstrapping as the initial warm-up of the network (or a new INM entity) where each node makes the initial contact with its INM neighbours. Note that the discovery also refers to the continuous process of maintaining the information updated (including the network status). The discovery of nodes and the network topology are issues already addressed in the literature. In most of the the proposed mechanisms, the nodes send broadcasting messages to all neighbor nodes to obtain information from the topology of the network. This process must be done in the discovery phase. In terms of bootstrapping mechanisms, their main goal is to detect and contact nodes in the network. Several bootstrapping mechanisms are found in the literature and are addressed to different types of networks such as, peer-to-peer, ad hoc, wireless mesh and sensors networks.

According to [7], for the complete bootstrapping mechanism, some requirements are in place:

- A distributed bottom-up algorithm that constructs a spanning;
- A resource discovery algorithm for efficient dissemination of local connectivity information;
- Synchronization protocol that guarantee the connectivity from local interactions;

The major challenge in the design of a bootstrapping mechanism is to ensure scalability and robustness, which becomes more complex and less efficient with the increasing number of INM entities entering and exiting the network. It is necessary to develop an efficient bootstrapping mechanism that contains these two requirements to be applied in the INM concept.

The second phase requires the dissemination of information between the INM entities to perform the management decisions. The dissemination of information in the network also contains a widely set of approaches in the literature. Some examples can use simple flooding [8], probability-based flooding [9], a minimum connected dominating set (MCDS) Based [10], location-based [11], epidemic-based [12] and cluster-based [13]. Other types of approaches are necessary to solve constraints in flooding mechanisms [14]. In our approach, it is required an efficient mechanism that distributes information between the nodes, considering the scope and type of information. A mechanism is also required to distribute local decisions to sets of nodes cooperating in the management process and to disseminate the results of the decision process. Beyond the communication process that defines to which INM entity specific information should be sent, it is very important to ensure the knowledge of information management, that is, which set of information should be distributed to which nodes.

In the third phase, it is required to define the communication process to disseminate the local management decisions to provide global cooperative decisions between the INM entities. Afterwards, it will be created primitives towards the optimized communication process between the INM entities.

The last phase provides the dissemination of the final decision that should be sent to the nodes in order to enforce it. It is required to define also which nodes need to receive the information to provide the required action, which nodes need to access the information, and how to identify them to optimize the dissemination process.

III. Related Work: Discovery and Bootstrapping Mechanisms

This section presents some relevant papers that discuss the subjects of discovery and bootstrapping in different types of networks.

A. Discovery Mechanisms

In [15], the authors abstract the problem of nodes discovery in a network with unknown size, using gossip method. The authors studied the problem using direct graphs, where each vertex represents the node participant and the edge represents the knowledge about other nodes. The nodes that participate in the network send gossip messages from their local information to other nodes. The authors still present four communication models *A*, *B*, *C*, *D* to discover nodes. In the models *A* and *B* they sent *multicast* messages with the size O ($n \log n$), while

TABLE I
COMPARISON BETWEEN DISCOVERY MECHANISMS

Works	Network Type	Mechanism	Drawbacks
[15]	Networks of unknown size	Gossip-Based Algorithms	Overhead of redundant messages
[16]	Sensor	Probabilistic Algorithms	Requires global synchronization of the network
[17]	P2P Systems	Hybrid (Multicast and Directory)	Do not work well when the network growing in number of nodes
[18]	Wireless Mesh and Ad Hoc	Probabilistic Algorithms	The performance degrades if the estimates of network size is inaccurate
[19]	Overlay	Gossip Protocol	This protocol spends overhead of synchronization
[20]	Ad Hoc	Analogy of ants behavior	Depends on specific software agents platforms

the other two models C and D have also *unicast* messages with O (n log n).

The problem of determining the neighbours in a wireless network is emphasized in [16]. The authors consider the problem of discovery of nodes in two different layers, physical and medium access. Therefore, the authors propose a neighbour distributed algorithm that allows each node into the network to have a complete or partial list of its neighbours.

The authors in [17] propose a hybrid peer discovery mechanism based on multicast and directory service. The hybrid mechanism proposed three components: *Directory*, *Multicast Advertisement* and *Central Cache*. The first is the main component, as it is responsible to provide a directory service containing peer and service list of all peer in the network. The peer list contains interface, directory port, local IP address and an external IP address. The service list contains the service names and indicates the participating peers in the peer list.

The problem of neighbour discovery in static wireless ad hoc networks is addressed in [18]. The authors propose two classes of probabilistic neighbour discovery algorithms, Direct-Discovery and Gossip-Based Algorithms. In both algorithms, any node must receive at least one successful transmission from its neighbours in order to discover those neighbours. The directed-discovery algorithm operates with least successful transitions to discover their neighbours.

In [19], the authors propose T-Man, a Gossip-based protocol, which builds the overlay topology in a distributed way. The strong points of this protocol is quickness, robustness, and simpleness to implement. However, this protocol needs a large amount of messages to synchronize the nodes topology. Thus, this becomes a strong drawback in a large network.

The bio-inspired approaches are other alternatives to discover nodes in dynamic networks. Many approaches are proposed in this field. There are some examples based on software agents for the discovery of the topology [21] [22] [20]. In [20], the authors use an analogy similar to the ant's behavior. The main idea is analogous to the ant's communication behavior (which uses pheromones) to exchange the information between the nodes in the network.

Table I summarizes the main characteristics and drawbacks of each described approach.

B. Bootstrapping Mechanisms

In *peer-to-peer* area, there are numerous mechanisms to provide bootstrapping, centralized and decentralized [14], such as, Content Addressable Network (CAN), Chord, Pastry, Tapestry as centralized, Multicast P2P and caching as hybrid, and Spanning Tree algorithms and Dynamic Domain Name System (DDNS)-Based Bootstrapping as distributed.

In [23], the authors emphasize that, when autonomic manager is attached into a running systems, it is required to have a bootstrapping mechanism. The main challenge involving the bootstrapping in autonomic manager is the need to have current and detailed state information, and then the need to minimize loading and element/systems management that provides the state information of Autonomic Manager (AM). To solve this challenge, the authors propose a bootstrapping state mechanism that provides a trade off between speed, state information acquisition, and loading in management system.

In [26], the authors propose an automate bootstrapping mechanism based on DDNS (Dynamic Domain Name System). This approach detects one existing peer-to-peer infrastructure and automatically joins the peer nodes into the network. The authors emphasize that the goal of bootstrapping mechanism is to find an already existent member of peer-to-peer system. Otherwise, without bootstrapping process, the peer-to-peer networks will work in a isolated way, reducing the overall system performance for all peers nodes in the network. In addition, the authors describe a system model that consists in a set of peers that are connected by a common communication network, as the Internet. In this system model, the peers are reliably sending messages and the peers join and leave the network without sending any further message. The main idea is to use DDNS-based bootstrapping mechanism to associate the IP address to an existence peer into the domain name already created.

In [25], the authors address the bootstrapping problem in mobile ad hoc networks. To solve this problem, the authors present a method of bootstrapping in P2P overlay networks running in the ad hoc networks environment. This method involves multicast P2P overlay joins queries and responses, and caches the results from all nodes. In addition, when one node desires to join a P2P overlay network, this node multicasts a network-wide join request. Next, when the join request is received, it will multicast a network-wide response. Finally, the main idea in this work is to apply the P2P techniques into mobile ad hoc networks.

In [7], the authors propose a self-organized bootstrapping mechanism to directional wireless networks. The main challenge in this type of networks is that the nodes have local connectivity information and limited number of transceivers. The authors propose a scalable bootstrapping model that contains three main requirements: (i) a distributed bottom-up algorithm that constructs a spanning, (ii) a resource discovery

TABLE II
COMPARISON BETWEEN BOOTSTRAPPING MECHANISMS

Works	Network Type	Mechanism	Scalability	Robustness	Distributed
[23]	Integrated Digital Enhanced Network (iDEN)	Bootstrapping state	no	moderate	Yes
[24]	Overlays	Peer Cache and Mediator-Based	moderate	moderate	No
[25]	MANET's	Multicast P2P and Caching	moderate	high	Hybrid
[7]	Directional Wireless Networks	Spanning tree algorithms	high	high	Yes

algorithm for efficient dissemination of local connectivity information, and (iii) a synchronization protocol that guarantees the connectivity from local interactions. In addition, the authors describe that a complete bootstrapping solution requires the integration of algorithms and protocols for determining connections, exchanging information between nodes and guaranteeing coordination and synchronization.

Table II summarizes the main characteristics of each described approach.

C. Final Considerations: Previous Works

After analyzing the related work, we argue that most uses a non-controlled or probabilistic flooding to perform the dissemination of information between nodes, which is less efficient with the increasing number of nodes in the network. Furthermore, most of the works miss the importance of bootstrapping as the initial warm-up of the network or specific new entity and according to the Table II, only [7] accomplishes bootstrapping main requirements, such as scalability and robustness in fully distributed way. In short, new approaches are necessary to efficiently bootstrap, discover and disseminate information in the network aiming at solving the existing drawbacks as presented in Table I. In order to overcome the limitations of previous works, we propose a new algorithm for bootstrapping, discovery and dissemination of information based on *Hide and Seek* [27] concept. The H&S algorithm uses probabilistic directional search to discover and disseminate proper information over the network.

IV. DISCOVERY AND BOOTSTRAPPING ALGORITHM: A HIDE AND SEEK CONCEPT

Network bootstrapping and discovery are two essential mechanisms to ensure the proper information dissemination in distributed In Network Management process. For this reason, the H&S algorithm was designed to ensure the communication of relevant and sufficient information on each In Network Management entity to ensure decision-making processes. Two roles were considered in the H&S algorithm: *seeker* and *hider*. A *seeker* node sends directional contact messages to the neighbourhood using a *probabilistic eyesight direction* (PED) function. This function aims to narrow the directions through which contact messages are sent. Once contacted, *hider* or *seeker* synchronize their knowledge, keeping track inside of the local repository of each other. When being contacted, the *hider* becomes a new *seeker* and the process is repeated until all nodes have been contacted. Our bootstrapping and discovery algorithm is based on the *Hide and Seek* game analogy, where the main goal is the same, which is to find all hidden players in a specific area. In the following paragraphs, we present the details of the algorithm steps.

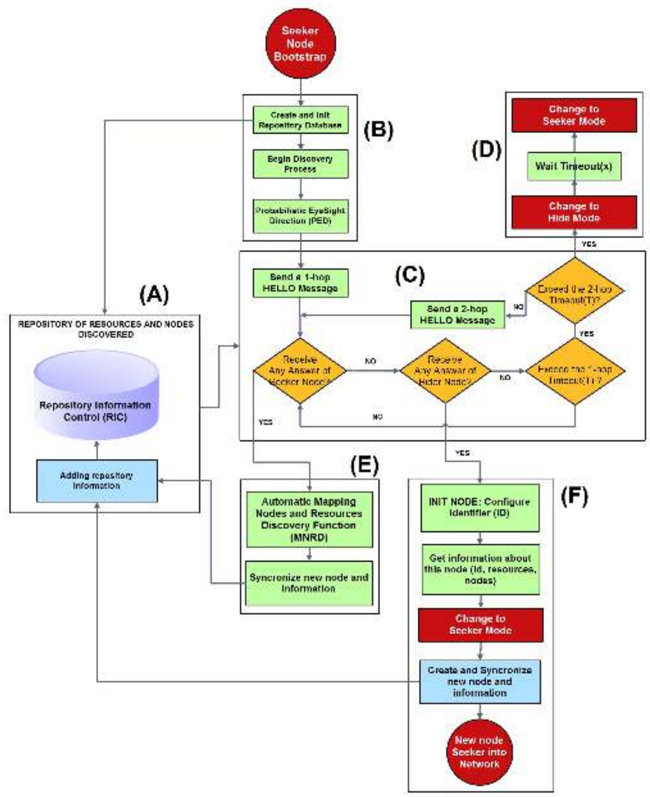

Fig. 2. H&S bootstrap process sequence steps

As depicted in the Fig. 2 , the discovery and bootstrapping algorithm starts when a new node (*seeker*) contacts another *seeker* or a *hider* node into the network. Note that Fig. 2 shows the view of one particular *seeker* node. Note also that, in the beginning of the algorithm, at least one node needs to work as a *seeker* to start the discovery process.

In step A.) the *seeker* node starts the algorithm with the creation of a repository. In this sense, each repository is created locally, and it is responsible for adding, updating and refreshing all gathered information during the discovery process. The *Repository Information Control* (RIC) controls the repository and is used to classify the type of information recorded, e.g resources available, network size-awareness, network domain diameters, etc. The RIC function guarantees integrity and readiness of information access.

The step B.) contains the creation and initialization of the *Init Repository Database*, the *Begin Discovery Process*, and the *Probabilistic Eyesight Direction (PED)*. The *Init Repository Database* process is responsible to initialize the local repository of each *seeker* node. Moreover, it records some local nodes information e.g (IP, MAC, etc.). The *Begin Discovery Process* initializes the discovery process algorithm and starts the process of discovery of nodes and information. The PED chooses the optimal direction of the search based on neighbours information of initial starting point (e.g, first node *seeker* to begin the search). We assume a starting point node *Sp* that has k neighbour nodes, e.g., k means the amount of wireless nodes that are in the surrounding area of *Sp*.

Let (Sh) be a neighbour vector which contains all neighbours of a starting poinbt node defined by

$$Sh = \sum_{i=1}^{k} \alpha_i \mid \alpha_i \in Sp, \alpha_i, 1 \leq i \leq k. \quad (1)$$

Where α_i represents the neighbours index of specific starting point *Sp* with limit of $1 \leq i \leq k$ discovered neighbours. Initially, the vector *Sh* is created using the discovered neighbours of initial starting point *Sp* vector.

In fact, we can define the PED as follows

$$PED \leftarrow P\gamma(Sh[\alpha_i], 1 \leq i \leq k) \quad (2)$$

Where $P\gamma$ represents the next choice of dissemination, based on the index α_i of neighbour. In addition, $P\gamma$ denotes the probability of choice $\sum P\gamma(i) = 1$, like the behavior of *seeker to hide* search, found in [27], where the goal is to find the *hider* nodes in a random matrix with *NxN* dimensions. The main goal of PED function is to choose based on Pγ probability, the optimal direction to ($Sh[\alpha_i]$) vector. This process is repeated recursively until all nodes on index $Sh[\alpha_i]$ have been contacted.

Step C.) proceeds like a contact between the neighborhood. This contact is realized using diameter messages of 1-hop or 2-hops. The direction contact is determined by the PED, and any *seeker* node can send a message of 1 or 2-hops and wait for an answer, e.g., of any *seeker* (in step (F)) or, *hider* (in step (F)). Moreover, when a *seeker* node does not receive any contacted messages of 1 or 2 hops before reaching a configured timeout, it executes the step (D).

Step D.) works in special cases when the *seeker* node has a inefficient search. For instance, when a *seeker* node waits a long time without receiving answers from any node, it becomes a *hider* node. Thus, the probability of another *seeker* to contact the *hider* node is higher. To sum up, this process avoids that the *seeker* nodes spend long time for the *hider* answers.

Step E.) proceeds when a *seeker* node receives a contact of another *seeker* node. When this happens, the contacted *seeker* begins the *Mapper Nodes and Resources Discovery* (MNRD) function, where it obtains specific information of each resource available in each *seeker*. After this process, the new information is synchronized on both *seeker* repositories. In addition, each *seeker* node has an internal identifier that performes node differentiation into the network.

Step F.) proceeds when *seeker* node receives a answer contact message of any *hider* node. After receiving the answer contact, the *seeker* node gets local information (resources available, etc), it synchronizes the information on repository between the *seekers* and change the status of *hider* to new *seeker* into the network. This step is complete when all *hider* nodes contacted becomes new *seeker* nodes. In addition, this is a preliminary stop condition of the algorithm.

V. Simulation

In this section we present the developed scenario used to evaluate the effectiveness of the proposed algorithm, we compare H&S with non-controlled flooding and probabilistic (gossip) flooding.

In order to evaluate the performance of H&S, we have implemented a simulation scenario in MATLAB v.7.0 R14, where several static nodes were randomly placed in a specific environment size (See Fig. 4 and 5). In addition, in this simulation, we considered the ideal network conditions, with no link corruption and with all nodes in the coverage area of themselves.

The Matlab simulation is capable of measuring the overhead of messages and amount of simulation cycles between the proposed approaches. In terms of simulation cycles, we record the amount of time that the recursive function takes to discover neighbor nodes. In terms of messages exchanged, we record the amount of links formed between the network nodes and the required messages to provide this linkage.

Depth First Search (DFS) gossip flooding [28] and non-controlled flooding algorithms were analyzed for comparison purposes. The metrics considered were the simulation cycles and the number of messages exchanged. We consider the node communication as a directional graph G = (S,H), where S is the *seeker* node and H is the *hider* node. The links between them are represented by (α, β and δ), where α represents the first contact message, β represents the first answer contact message and δ represents the synchronization message between *seeker* nodes, as depicted in Fig. 3 . The nodes are placed according to the randomness function, where N = rand(1, 100) is a randomness matrix *(N)* of 1 to 100 nodes. Initially we consider a starting point *seeker* node to begin the discovery. To compare the same scenario in different algorithms, the same randomness *seed* and distance unit was used. Different scenarios used different seeds.

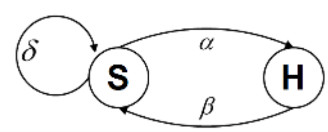

Fig. 3. H&S directional nodes communication

These scenarios represent H&S algorithm (Fig. 4) and DFS (Depth First Search) Gossip flooding (Fig. 5). Both scenarios were simulated using the same amount of nodes disposal inside both environments. In addition, each link or (line) represents

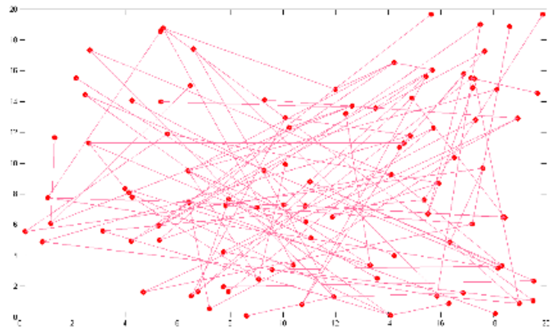

Fig. 4. H&S scenario with 100 nodes disposal in a specific environment

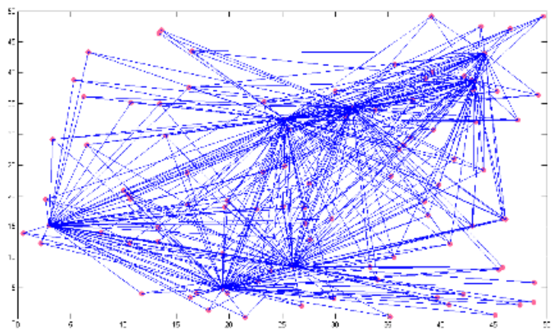

Fig. 5. DFS (Gossip) scenario with 100 nodes disposal in a specific environment

the process of contact, collect and synchronize information between the nodes.

In our simulation some parameters can be changed, such as:

- Coverage Area (txRange): Configures the coverage area of all nodes in the specific environment. This is an integer value e.g. 15. The Equation 3 is used to calculate the distance matrix between all nodes.

$$Dm[i,j] = \sqrt{((XL[i]) - (XL[j])^2 + (YL[i]) - (YL[j])^2)} \quad (3)$$

  Where, *XL* and *YL* represent the coordinate points of adjacent matrix, and *Dm* represents the adjacent distance matrix.
- Number of Nodes (NumNodes): Configures the number of nodes in the network. This is an integer value e.g. 100.
- Environment Size (EnvSize): Configures the proportional size of environment. This parameter can vary in the interval [*n*...*n-1*]. This is an integer value e.g. 30.

The main purpose of this simulation is to evaluate, in terms of communication, the abovementioned approaches in dense environments, e.g. we vary the number of nodes from 10 to 100, and evaluate the behavior of communication process between the nodes in each particular case. In order to prove of concepts, the following metrics are considered, *number of necessary messages*, that describes the number of necessary messages to contact the node and synchronize its information and, *the number of simulation cycles*, that represents the amount of recursive steps to discover all nodes in the network.

VI. Preliminary Results

This section presents the obtained results in our simulated scenarios. We measured the efficiency of H&S algorithm in terms of overhead of messages needed to acquire nodes information and synchronization, and the number of required simulations cycles to discover nodes information. The results proved the following conclusions:

- As depicted in Fig. 6 , our H&S algorithm has lower overhead of messages to contact, collect and synchronize nodes information when compared to the probabilistic and non-controlled flooding approaches, while the nodes increase in the network. Moreover, H&S is more scalable for large and dense scenarios than the probabilistic gossip flooding.
- As show in Fig. 7 , the H&S is more efficient than probabilistic gossip and non-controlled flooding to discover nodes information, due the fact that the H&S needs lower simulation cycles to discover all neighbours nodes in its cover area.
- As depicted in Fig. 4 and Fig. 5 , the H&S scenario requires a lower number of links between the nodes when compared to probabilistic flooding (gossip). This is desirable in large scale environments.

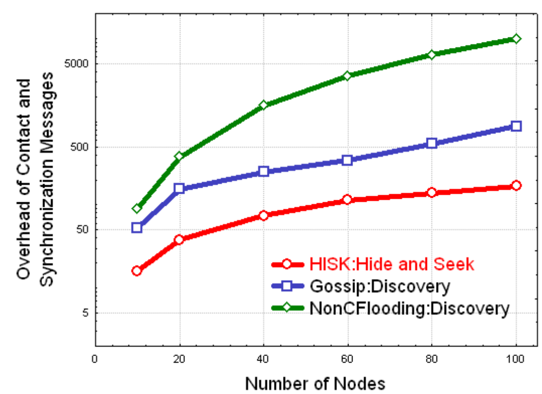

Fig. 6. Evaluation of number of contact and sync messages with 100 nodes

To validate the results, we consider 95% confidence intervals generated from the simulation graphs. Although not shown in the figures, the confidence intervals are very small. As an example, for 80 nodes, the confidence values are a small percentage of the mean values: 0.78% for Hide and Seek, 1,33% for gossip probabilistic flooding, and 1,59% for non-controlled flodding, when considering the number of simulation cycles. For the number of messages exchanged, these values are around 5%.

VII. Conclusions and Future Work

In this paper we introduced the importance of In Network Management concept and its phases, in terms of communication, to ensure INM paradigm to be a reality in future

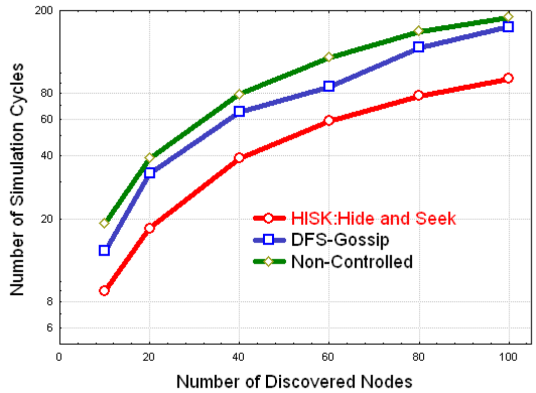

Fig. 7. Evaluation of simulation cycles with 100 nodes

networks. We then proposed a new H&S algorithm to discover network information and synchronization, with small overhead and number of exchanged messages. Through MATLAB simulations, we showed that the efficiency of our H&S in terms of overhead of messages and number of cycles to discover nodes information, is significantly increased when compared to gossip and non-controlled flooding mechanisms. We argue that the use of the PED function provides better scalability, robustness support and improves efficiency of the search mechanism when compared to the other approaches.

As future work, we plan to implement our H&S algorithm in *OPNET Modeler* simulator and evaluate its behavior on network virtualization scenarios. We will also adapt and enhance the algorithm to provide the dissemination of local and global decisions in the INM paradigm.

REFERENCES

[1] B. Jennings, S. van der Meer, S. Balasubramaniam, D. Botvich, M. Foghlu, W. Donnelly, and J. Strassner, "Towards autonomic management of communications networks," *Communications Magazine, IEEE*, vol. 45, no. 10, pp. 112–121, October 2007.

[2] A. G. Prieto, D. Dudkowski, C. Meirosu, C. Mingardi, G. Nunzi, M. Brunner, , and R. Stadler, "Decentralized in-network management for the future internet," *International Workshop on the Network of the Future, in conjunction with IEEE ICC 2009, Dresden, Germany, June 2009*, 2009.

[3] C. Foley, S. Balasubramaniam, E. Power, M. P. Leon, D. Botvich, D. Dudkowski, G. Nunzi, and C. Mingardi, "A framework for in-network management in heterogeneous future communication networks," in *MACE '08: Proceedings of the 3rd IEEE international workshop on Modelling Autonomic Communications Environments*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 14–25.

[4] M. Brunner, D. Dudkowski, C. Mingardi, and G. Nunzi, "Probabilistic decentralized network management," in *Integrated Network Management, 2009. IM '09. IFIP/IEEE International Symposium on*, June 2009, pp. 25–32.

[5] N. Niebert, S. Baucke, I. E. Khayat, M. Johnsson, B. Ohlman, H. Abramowicz, K. Wuenstel, H. Woesner, J. Quittek, and L. M. Correia, "The way 4ward to the creation of a future internet." in *PIMRC*. IEEE, 2008, pp. 1–5. [Online]. Available: ´´

[6] W. Project, "The fp7 4ward project." [Online]. Available: http://www.4ward-project.eu/

[7] S. Milner, J. Llorca, A. Anibha, and U. Vishkin, "A bootstrapping model for directional wireless networks," vol. 10, no. 12, pp. 840–842, December 2006.

[8] C. Chi, D. Huang, D. Lee, and X. Sun, "Lazy flooding: A new technique for information dissemination in distributed network systems," vol. 15, no. 1, pp. 80–92, Feb. 2007.

[9] V. Drabkin, R. Friedman, G. Kliot, and M. Segal, "Rapid: Reliable probabilistic dissemination in wireless ad-hoc networks," in *Proc. 26th IEEE International Symposium on Reliable Distributed Systems SRDS 2007*, 10–12 Oct. 2007, pp. 13–22.

[10] N. Meghanathan, "An algorithm to determine the sequence of stable connected dominating sets in mobile ad hoc networks," in *Proc. International Conference on Internet and Web Applications and Services/Advanced International Conference on Telecommunications AICT-ICIW '06*, 19–25 Feb. 2006, pp. 32–32.

[11] H. Sabbineni and K. Chakrabarty, "Location-aided flooding: an energy-efficient data dissemination protocol for wireless-sensor networks," vol. 54, no. 1, pp. 36–46, Jan 2005.

[12] *EraMobile: Epidemic-Based Reliable and Adaptive Multicast for MANETs*, March 2007.

[13] C. Chen, J. Ma, and J. Salomaa, "Simulation study of cluster based data dissemination for wireless sensor networks with mobile sinks," in *Proc. 10th International Conference on Advanced Communication Technology ICACT 2008*, vol. 1, 17–20 Feb. 2008, pp. 231–236.

[14] J. K. W. R. Heinzelman and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking*, Seattle, WA USA, 1999, pp. 174–185.

[15] K. M. Konwar, D. Kowalski, and A. A. Shvartsman, "Node discovery in networks," *Journal of Parallel and Distributed Computing*, vol. 69, no. 4, pp. 337 – 348, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B6WKJ-4V3HF9B-2/2/d4b1d66f80f6e3e62a69d91ed7da93ff

[16] *An asynchronous neighbor discovery algorithm for wireless sensor networks*, vol. 5, no. 7, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/B7576-4K1X4F3-1/2/e460365e55c40122c67695f6cc947c1a

[17] Y. Liu, G. Zhu, and H. Yin, "A practical hybrid mechanism for peer discovery," in *Proc. International Symposium on Intelligent Signal Processing and Communication Systems ISPACS 2007*, Nov. 28 2007–Dec. 1 2007, pp. 706–709.

[18] S. Vasudevan, J. Kurose, and D. Towsley, "On neighbor discovery in wireless networks with directional antennas," in *Proc. IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2005*, vol. 4, 13–17 March 2005, pp. 2502–2512.

[19] M. Jelasity, A. Montresor, and O. Babaoglu, "T-man: Gossip-based fast overlay topology construction," *Computer Networks*, vol. 53, no. 13, pp. 2321 – 2339, 2009, gossiping in Distributed Systems. [Online]. Available: http://www.sciencedirect.com/science/article/B6VRG-4VYXMV4-1/2/5734dd4ac83cdea78933601692a7b5a1

[20] B. T. Nassu, T. Nanya, and E. P. Duarte, "Topology discovery in dynamic and decentralized networks with mobile agents and swarm intelligence," in *Proc. Seventh International Conference on Intelligent Systems Design and Applications ISDA 2007*, 20–24 Oct. 2007, pp. 685–690.

[21] R. R. Chpudhury, S. Bandyopadhyay, and K. Paul, "A distributed mechanism for topology discovery in ad hoc wireless networks using mobile agents," in *Proc. MobiHOC Mobile and Ad Hoc Networking and Computing 2000 First Annual Workshop on*, 11 Aug. 2000, pp. 145–146.

[22] N. Migas, W. J. Buchanan, and K. A. McArtney, "Mobile agents for routing, topology discovery, and automatic network reconfiguration in ad-hoc networks," in *Proc. 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems*, 7–10 April 2003, pp. 200–206.

[23] G. W. Cox, W. L. Johnson, J. Strassner, and D. Raymer, "Bootstrapping device state in managed systems," in *Proc. IEEE Network Operations and Management Symposium NOMS 2008*, 7–11 April 2008, pp. 767–770.

[24] M. Knoll, A. Wacker, G. Schiele, and T. Weis, "Decentralized bootstrapping in pervasive applications," in *Proc. Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops PerCom Workshops '07*, 19–23 March 2007, pp. 589–592.

[25] A. Mawji and H. S. Hassanein, "Bootstrapping p2p overlays in manets," in *Proc. IEEE Global Telecommunications Conference IEEE GLOBECOM 2008*, Nov. 30 2008–Dec. 4 2008, pp. 1–5.

[26] M. Knoll, A. Wacker, G. Schiele, and T. Weis, "Bootstrapping in peer-to-peer systems," in *Proc. 14th IEEE International Conference on Parallel and Distributed Systems ICPADS '08*, 8–10 Dec. 2008, pp. 271–278.

[27] M. M. Flood, "The hide and seek game of von neumann," *Management Science, Vol. 18, No. 5, Theory Series, Part 2, Game Theory and Gaming*, vol. 18, no. 5, pp. P107–P109, January 1972.

[28] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 2, March 2003, pp. 1124–1130 vol.2.