# Nodes Discovery in the In-Network Management Communication Framework

Lucas Guardalben, Tomé Gomes, António Pinho,
Paulo Salvador, and Susana Sargento

Instituto de Telecomunicações, University of Aveiro, Portugal
{guardalben,tomegomes,antonio,salvador,susana}@ua.pt

**Abstract.** The main role of a communication framework in distributed autonomic management is to support the dissemination of management information between network nodes. In distributed autonomic management, each network node intelligently self-adapts its behavior through collaboration and cooperation between the several nodes. In this paper, we propose a set of communication mechanisms between self-managed network nodes, comprehending the several stages of communication, including a bootstrapping, discovery and election of entities, and ensure the base of communication of information between nodes to perform the collaborative decisions and to enforce these decisions. We propose a bootstrapping and discovery mechanism that uses the concept of *Hide & Seek*, where the entities change their role dynamically according to events in the network, with dynamic probing intervals according to the number of *Seekers* entering or leaving the network. We compare our discovery approach with current solutions, and we show that our mechanism is more efficient both in terms of control messages overhead and convergence time.

**Keywords:** Communication Framework, In-Network Management, Bootstrapping, Discovery.

## 1 Introduction

Over the last decade, the most widespread approaches for traditional management were the Simple Network Management Protocol (SNMP) [7] and Common Management Information Protocol (CMIP) [32]. For emerging dynamic and large-scale networking environments, as envisioned in Next Generation Networks (NGN), it is expected an exponential growth in the number of network devices, including the widespread of mobile devices. In large-scale environments, these management approaches have serious problems in terms of scalability, due to their centralized management characteristics. The works in [3,8] refer the need for a distributed management approach for increased scalability. However, these approaches require the need for collaboration and cooperation between network entities. Moreover, the high level of management automation tasks is an important requirement for new network management approaches. The increasing popularity of autonomic and self-management concepts raised several challenges

K. Pentikousis et al. (Eds.): MONAMI 2011, LNICST 97, pp. 145–157, 2012.

and opportunities regarding the management of NGNs [11,5,29], to optimize the management and automatic reaction to network events.

An important requirement over distributed network management paradigm is the support for communication between entities for collaboration and cooperation management decisions. This type of collaboration and cooperation usually requires too much overhead to collect, synchronize and disseminate management information and decisions in the whole network. In response to this challenge, we propose a set of communication mechanisms, comprehending the several stages of communication, including a bootstrapping, discovery and election of entities, and ensure the base of communication of information between nodes to perform the collaborative decisions and to enforce these decisions. The work described in this paper addresses the lightweight communication and collaboration between the network entities in the bootstrapping, discovery and election processes. The proposed bootstrapping and discovery mechanism uses the concept of *Hide & Seek*, where the entities change their role dynamically according to events in the network, with dynamic probing intervals according to the number of *Seekers* entering or leaving the network. We compare our discovery approach (In-Network Management - INM) with three discovery solutions, Cisco Discovery Protocol (CDP) [10], *Overlook FING* [23], and the discovery approach of Open Shortest Path First (OSPF) [21]. We demonstrate that our discovery mechanism INM is more efficient both in terms of control messages overhead and convergence time.

The paper is organized as follows. Section 2 presents related work, and highlights our contribution with respect to previous work. Section 3 introduces the distributed In-Network Management paradigm and the communication framework. Section 4 describes the bootstrapping, discovery and election mechanisms proposed. Section 5 depicts the discovery mechanism results and their comparison with current approaches. Finally, Section 6 concludes the paper.

## 2 Related Work

Our related work investigates group communication frameworks for distributed network management and discovery approaches.

The group communication problem in distributed management systems is theme of research on [24]. The essential aim is to provide a lightweight communication infrastructure to decrease the overhead, reducing the number of extra messages for communication. A framework using IP plus SNMP for group communication is proposed in [28,24]. The main idea of this framework is to develop autonomous SNMP agents using IP multicasting. However, it lacks the flexibility of multicast group re-configuration according to applications and network demand. In [19], a reliable framework for group communication uses the hierarchy of servers and logical timestamps to ensure reliability and correct ordering of the group delivery. This framework uses unicast connections to emulate group communication, limiting significantly the performance robustness and scalability. Another group communication framework was proposed in [24] [2] which also lacks the multi-domain management support, which is an important requirement posed by NGNs. The previous mentioned frameworks are not able to cope with current demands and support for large-scale management.

Discovery can be described as a process where each node becomes aware of its surrounding neighbors presence. This process includes assessing quality of links/signals and providing information to identify the better path to the destination. In the last few years, a massive amount of approaches were proposed for discovering nodes/topology in sensor networks [17,31] as well as in ad hoc networks [25,15]; its need is directly influenced by the dynamic topology of their networks. The discovery can be performed at the Link-Layer, which discovers the physical topology at 1-hop [27,18], or at the IP-Layer, which collects information to determine the logical topology [20,16]. In terms of nodes/topology discovery, there are several solutions addressed in the literature; the most relevant are Asynchronous Discovery [4,13], Bio-Inspired [22,9], Directional Antennas [30], Hybrid-Peer Discovery [20], Probabilistic Discovery [18,9] and Beacon Assisted [27]. In general, these techniques send broadcast messages to all neighbor nodes to obtain information from the neighbors and topology from the network. However, all these mechanisms suffer from large overhead. On the wired networks side, routing protocols integrate the discovery functionalities in order to create a topology list. OSPF [21] is an adaptive routing protocol that uses Link State Advertisements (LSAs) packet types for neighbor discovery. Regarding to commercial discovery approaches, Cisco Discovery Protocol (CDP) [10] is a media- and protocol-independent mechanism that runs on all Cisco-manufactured equipment; *Overlook* FING [23] is another commercial example solution for nodes discovery, that uses broadcast ARP-messages to discover the nodes in the network. We argue that these discovery approaches all present large overhead in the discovery process, and that they cannot be used efficiently in the management communication framework.

## 3 In-Network Management Communication Framework

In this section, we present an overview of the In-Network management concept and of our communication framework.

### 3.1 In-Network Management: Overview and Requirements

In-Network Management (INM) [26] is a new paradigm, also studied on the scope of 4WARD project [1], that considers the support of management functionalities by the means of a fully distributed architecture, designing management functionalities inside the network elements. Thus, each network element has the capability to take decisions based on the knowledge obtained from the other elements. This approach requires continuous interactivity between entities in order to exchange information about each entity (and therefore the network). This information will allow the network to make automatic decisions, through collaboration between the network nodes, reacting to network changes (such as link failures, load variations, etc) and continuously optimizing the network resources.

As depicted in the Fig.1, a comparison between network management approaches is presented. In the traditional network management, the administrator of the network has the central control of management decision, interacting with the network management through the management commands. In self-management approaches, the control and decisions are subject to the control-loop in an automatic way. Therefore, most of the self-management approaches use centralized servers to control, act and disseminate the policies and rules. However, this external server approach turned out to be inadequate in terms of scalability.

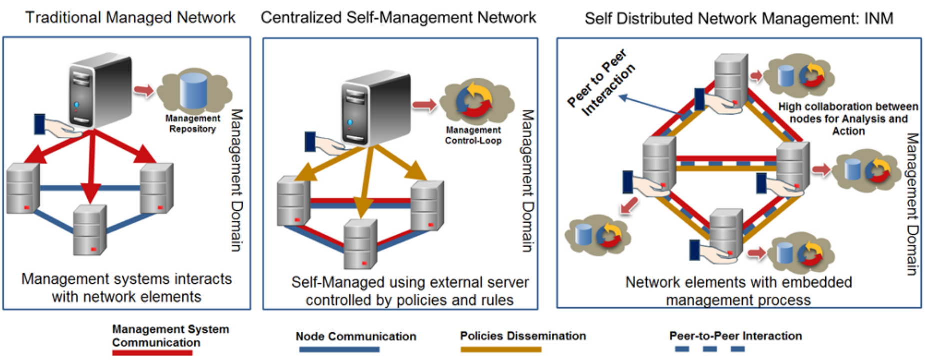


**Fig. 1.** INM-Comparison

As opposed to the traditional management and external self-management dedicated control, in the INM concept, each entity interacts with its peers and has the ability to take decisions based on the knowledge from the other elements, forming a network of collaboration and cooperation between entities [26]. The goal of INM is to achieve scalable and low complexity management for large-scale and dynamic network environments. The guiding principles for achieving this goal are the decentralization and self-organization. In order to achieve these goals a number of functional requirements were proposed [12], and we identified the most important in a distributed management communication infrastructure:

- Situation awareness: suitable mechanisms for real-time monitoring of network-wide metrics, group size estimation, bootstrapping, nodes and topology discovery, data search and anomaly detection.
- Scalability: support scalability in terms of network size, e.g. the number of network components to be managed; it must provide mechanisms to aggregate the network in domains or in federated multi-domains.
- Functional Comprehensiveness: provide functional richness to support a variety of essential management tasks.
- Extensibility: assure that capabilities of nodes can be extended with new functionalities.
- Small Footprint: with respect to storage space, bandwidth consumption, energy consumption, and other resources.

In the next sub-section we introduce our framework for communication, highlighting all phases and interactions.

### 3.2 Framework for Communication

Fig.2 depicts the proposed communication framework. It uses a communication infrastructure and also peer-to-peer interaction between INM entities: each entity needs to have functionalities to start the management process by itself, and contact the neighbors to initially acquire information.

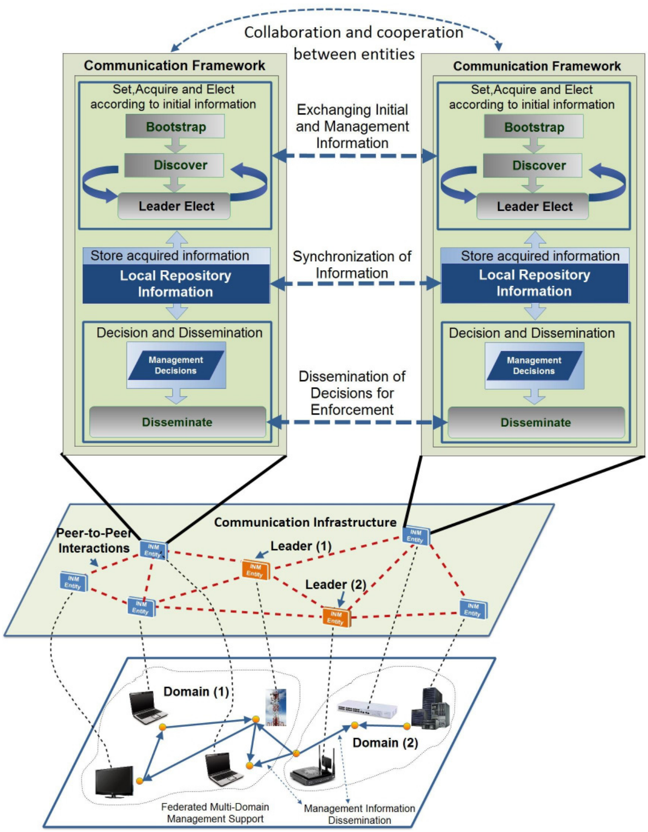


**Fig. 2.** Communication Framework for In-Network Management

This information will be stored in local repositories of each node and will be exchanged between the surrounding neighboring entities in order to establish a high level knowledge information repository. The communication framework is divided in three important phases: exchange of initial information (bootstrapping and discovery), synchronization of information between network nodes, and dissemination of local management decisions and enforcement.

In the first phase, the bootstrapping is the initial warm-up of the network (or a new INM entity), where each INM entity makes the initial contact with its INM entity neighbors. Note that the discovery also refers to the continuous process of maintaining the information updated (including the network status). In this phase, it is required to exchange initial information in order to acquire the primary contact information. After obtaining this initial information, the INM entities are able to decide on which node in each region or community is going to become the leader at that time. We consider that each INM region contains a leader for intra/inter domain communication between INM entities, and for dissemination of management information and decisions to enforcement. If a new entity enters in the network with better characteristics than the actual leader, this entity will become the leader of the group. Similar process exists when the characteristics of the entities change. This process is, therefore, dynamic and dependent on the actual nodes and their characteristics.

The second phase performs the process of acquiring and exchanging management information between the entities to perform the synchronization of information acquired between INM entities. We are planning to divide the acquired information in the initial information and management information. The initial information will be collected and stored in local node tables, and then these tables will serve as a base for preliminary management decision process using incomplete network information. Regarding to management information, the success of management decisions are correlated to the accuracy of the information collected and synchronized. Reliable responses are the key aspect of this phase, and an alternative to ensure the reliable information is to use real-time databases [6]. Real-time databases are complemented by intelligent algorithms for synchronizing and updating the information. Also, real-time databases can separate the information on levels, which will be easier to understand, since the relevant management information can be used at the moment of a decision.

In the third phase, it is required to disseminate the local management decisions in order to provide global cooperative decisions between the INM entities. Afterwards, primitives towards the optimized communication process between the INM entities will be created. In this phase, it is also disseminated the final decision that should be sent in order to enforce the management decision. It is required to define also which entities need to receive the information to provide the required action, as well as how to identify them to optimize the dissemination process.

The federated multi-domain support will also be investigated, taking into account the different management approaches in each domain.

## 4 Bootstrapping, Discovery and Election: A Closer Look

Network bootstrapping, discovery and election are three essential mechanisms to ensure the initial information dissemination in our distributed management infrastructure. Bootstrapping corresponds to the initial warm-up of the network (or new entity), where static properties are learned by each INM entity (e.g. local resource capabilities or topology). Discovery refers to the continuous process of maintaining the discovered information updated, while the election is the procedure of choosing the best quality entity amongst several others, to perform special actions, such as the dissemination of management information and of the management decisions. In this study, we consider the bootstrapping and discovery in wired networks, such as the backbone of network operators.

These mechanisms have strong correlation in our framework: when a new entity enters in the communication infrastructure, the bootstrapping process configures initial information (e.g identifiers, timers, local repository functions, etc). After that, the neighbor discovery is started, followed by the election process. For discovery, we propose an extended version of Hide and Seek (H&S) mechanism [14] and two roles are considered: INM_Seeker and INM_Hider. In order to create and gather information of surrounding neighbors, an INM_Seeker entity sends multicast HELLO contact messages to its neighborhood using a defined Time-to-live (TTL). In addition, all gathered information is recorded in a local partial view of each INM_Seeker. According to Fig.3, the entities exchange HELLO messages using TTL 1, for example, to avoid long cycle messages. Notice that we consider that each entity does not need to known the entire network. In our proposal, the HELLO interval is adaptive and can increase or decrease according to the number of entities that are present in each INM_Seeker partial view. This approach avoids the extra overhead of synchronization messages due to the cooperation and collaboration between the seekers. The partial view is a local table that records initial information, in terms of identifiers (MAC and Internal identifier), source and destination IP addresses and roles (seeker or hider). In order to calculate this adaptive HELLO interval, it is set a *Max_interval* at the bootstrapping process, and the initial value calculated will be a random between 1 to *Max_interval* and the number of *INM_Seekers* evolved is represented by *Nseekers*. The adaptive HELLO is calculated according to the given equation.

$$Hello_{Interval} = \left( \frac{Random(Max\_interval) * Nseekers}{Max\_interval} \right) \tag{1}$$

This random number generated is automatically adjusted according to the amount of INM_Seekers gathered in each partial view. In the end, this process will be dynamically adjusted each time a HELLO contact message is sent.

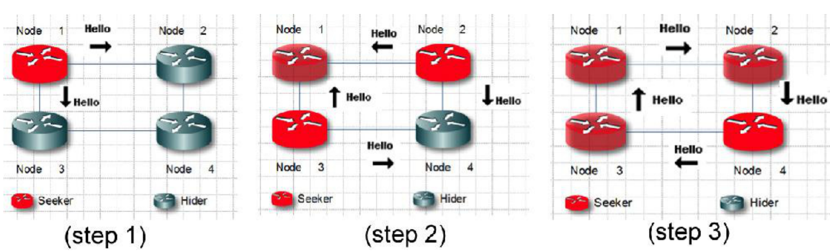


**Fig. 3.** INM-Discovery role interaction in three steps

The *INM_hider* waits a *INM_Seeker* contact message, and then becomes a new seeker that starts the discovery of hider nodes, and the process is repeated until all entities have been contacted. With regard to the complexity avoidance of constructing and managing the INM entities, we develop all interactions between bootstrapping, discovery and election based on an automated process. This process creates, exchanges and sets up the INM entities dynamically without involvement of the administrators. This idea significantly facilitates the administration of the group communication infrastructure.

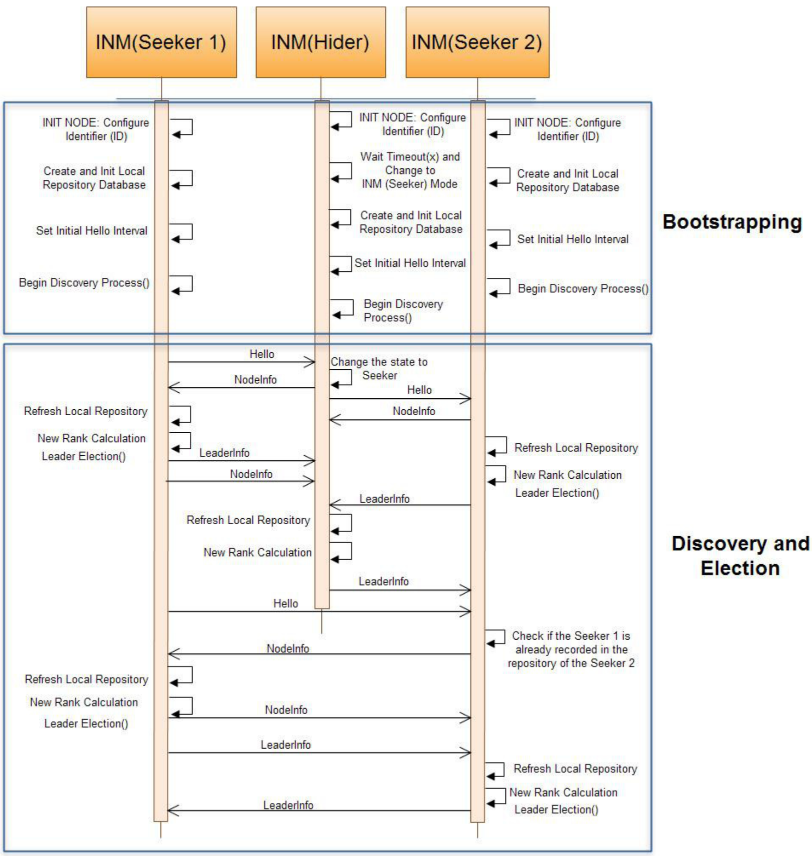


**Fig. 4.** Automatic INM entities' bootstrap, discover and election signaling process

In Fig.4 we demonstrate the automatic signaling process between the entities, starting with *INM_Seeker* or *INM_Hider* interaction.

### 4.1 Bootstrapping

When a new entity (seeker or hider) enters in the network, it configures a local identifier and initializes the local repository, and then, the discovery function is called. In the case of a hider, it waits for a random time (e.g 1 to 60 seconds) and if no *INM_Seeker* gets into contact , it changes its role to seeker and initiates the discovery process. We created an internal identifier that controls each entity, and it is composed by MAC address plus a random number (e.g 00:45:fa:54:a4-568945). Each entity sets this internal identifier in the bootstrapping process.

### 4.2 Discovery and Election

After the bootstrapping process calls the discovery procedure, the *INM_Seeker* sends a HELLO message containing (Msgtype, Hello ID) and waits for a NodeInfo response message containing (Msgtype, Hello ID, nNodes, Interface, pFreeRam, pFreeCPU, bandwidth, nIterfaces, type) of the contacted entity. Our mechanism scales well with network sizes in terms of message overhead, and the effect of this communication is minimal. This fact is explained taking into account the collaboration of each *INM_Seeker* through its partial views.

It is important to mention that, when multicast groups are formed, the messages are immediately propagated to the rest of the *INM_Seekers* in order to refresh the actual information between them. In the case of *INM_Seeker* to *INM_Seeker* communication, both of them check if the contacted seeker is already recorded in the local repository and then, synchronizes their local repository. For the group communication, we have IPv6 and multicast support. Moreover, each INM entity has the capability to store in a local repository all collected information about the surrounding neighbors. This local repository stores basically the source and destination IP Addresses, % Memory and free CPU, entity network interfaces, MAC addresses, Round-Trip-Time (RTT) for the contacted neighbours, the partial view and the links' bandwidth.

After a node gathers knowledge about the neighborhood, it is needed to decide which one is considered to be the best, and therefore the leader. Once a node enters the election phase, it will perform a rank calculation taking into account the above mentioned information. The rank is created according to the given equation.

$$Rank\_(n) = (w_1 * Bw + w_2 * (Free_{mem} + Free_{cpu} + RTT) + w_3 * (N_n + N_i)) \quad (2)$$

Where $Bw$ is the bandwidth (e.g 1.0 Mbps) of the link plus nodes' resources, $Free_{mem}$ is free memory and $Free_{cpu}$ is free CPU of a node $n$ (e.g $0.85 \approx 0.93$ in percentage), and $RTT$ is Round-Trip-Time calculated in pairs among the nodes already discovered (e.g between node 1 and node 2 is $\approx 0.479$ msec). $N_n$ and $N_i$ are the number of surrounding nodes and connected interfaces respectively (e.g node 1 has 3 nodes connected on eth0, eth1, eth2 communication interfaces). $w1$, $w2$ and $w3$ define the weights of the rank function. The node that has the maximum Rank, $max(Rank\_(n))$, is the leader. In order to avoid the consensus problem in this distributed system, each entity, according to prior gathered information, sends its local rank calculation to the highest ID plus interfaces node, to count the number of the repetition ranks. After that, each entity is informed about the leader of the group through a message that contains the ID of the chosen leader as well as the actual rank calculated. This process is continuous, until a new leader needs to be elected, according to those characteristics.

## 5 Results of the Discovery Process

The testbed scenario consists in a $5x5$ Grid, resulting in 25 machines with 512Mb RAM/558Mhz CPU and 1Gb storage, running Debian Lenny 2.6.26 as operating system. These machines are built through virtualization and are deployed in a 2x Intel Xeon 2.40Ghz (8 cores) server with 24Gb of RAM and 15000 RPM SAS disks running Centos 2.6.18 as operating system with Xen Linux Kernel v3.4. Our INM discovery protocol was implemented in C/C++ language, with IPv6 and multicast group support. With respect to the virtual links communication, the end-to-end link delay (e.g from entity 1 to 24) is 0.745 msec, and the bandwidth was prior configured to 1.0 Mbps for all virtual links.

In the evaluation study, we compared our proposed INM discovery against CDP, FING and the discovery approach of OSPF, since we are addressing wired networks. We run each bootstrapping and discovery mechanisms during an observation period of 60 seconds, with 5 independent runs. The presented results are the mean of 5 independent runs with a 90% confidence interval. Notice that the election mechanism is not evaluated in this study.

For each run, we analyzed the convergence time for the discovery and the overhead impact. The convergence time represents the required time to find all nodes in the network, while the overhead is the percentage of discovery-related packets in the overall traffic. In all experiments and studied mechanisms, the amount of traffic in the network that does not represent the discovery traffic is always the same, approximately $900 \approx 950$ packets.

CDP, FING and OSPF-Discovery work with fixed HELLO intervals. We used different values of HELLO interval (1, 5, 10 and 20 seconds) to assess the impact on both convergence time and overhead. In the INM discovery, this HELLO interval is set (5 sec) during the bootstrapping. After that, the interval is dynamically adjusted, according to the number of contacted *INM_Seekers* and *INM_Hiders*. In addition, the number of initial *INM_Seekers* and *INM_Hiders* was randomly configured, considering 50% of *INM_Hiders* and 50% of *INM_Seekers*.

Fig.5 compares the convergence time of all mentioned approaches. For 1 and 5 seconds of HELLO interval, FING, CPD and OSPF-Discovery perform better than the INM discovery protocol. However, the convergence time of INM remains the same while others increase for higher values of HELLO interval. We also changed the HELLO Adaptive Interval of the INM to be 20 seconds, and we obtained approximately the same convergence time as being it 5 seconds.

In Fig.6 it is shown the percentage of overhead. It is clear that INM has the lower overhead, even for the cases of high HELLO intervals. The overhead of the CDP and OSPF-Discovery decreases, while the overhead in INM and FING remains nearly constant, although FING's overhead is much higher. This low overhead of INM is obtained due to the role-based characteristic (where each

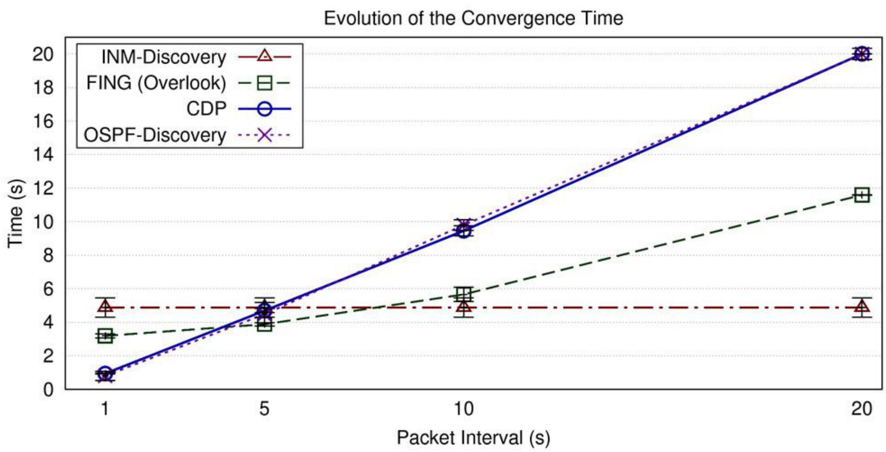


**Fig. 5.** Convergence Time varying the fixed HELLO interval of FING, CDP and OSPF-Discovery

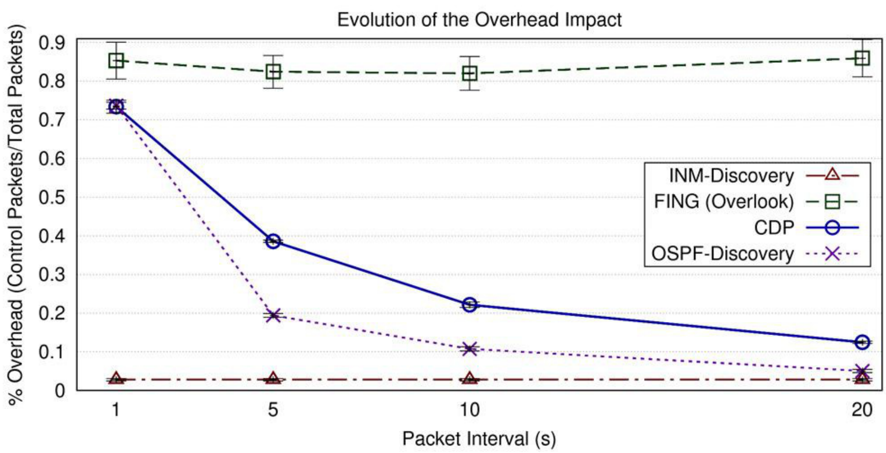


**Fig. 6.** Overhead Impact varying the fixed HELLO interval of FING, CDP and OSPF-Discovery

event is trigger-based) of our algorithm, besides the collaboration between nodes through the partial views.

## 6 Conclusions

This paper proposed a communication framework for distributed network management, comprising the bootstrapping and discovery processes. The most important features of our proposed approach are the fact that: (1) entities may change their role dynamically according to events or situations in the network; (2) it contains adaptive HELLO intervals in accordance to the amount of *INM_Seekers* that enter or leave the network; (3) events and triggers are executed in accordance to type of the entity contacted. We proved that our INM discovery mechanism is more efficient when compared to CDP, FING and OSPF-Discovery solutions, both in terms of discovery and convergence time and overhead impact in the network.

As future work, we plan to address the remaining phases of our communication framework: synchronization of management information, dissemination of local management decisions, and federated inter-domain plus network-wide metrics as well. In the wireless networks side, we plan to improve the leader election process using social-metrics.

**Acknowledgements.** This work was supported by the Fundação para Ciência e Tecnologia - FCT, through the grant SFRH/BD/62511/2009 and Portugal Telecom Inovação company, through the PANORAMA project ADI/QREN 3144. The authors also thank the partial support provided by UBIQUIMESH project PTDC/EEA-TEL/105472/2008.

## References

1. 4WARD: 4ward project (2008), `http://www.4ward-project.eu/`
2. Amir, E., et al.: Group communication as an infrastructure for distributed system management. In: Proceedings of Third International Workshop on Services in Distributed and Networked Environments, pp. 84–91 (June 1996)
3. Anderson, J.M., et al.: Distributed network management in an internet environment, November 3-8, vol. 1, pp. 180–184 (1997)
4. Borbash, S.A., et al.: An asynchronous neighbor discovery algorithm for wireless sensor networks, vol. 5, pp. 998–1016 (2007)
5. Brunner, M., et al.: Probabilistic decentralized network management, June 1-5, pp. 25–32 (2009)
6. Buchmann, A.: Real time database systems. In: Encyclopedia of Database Technologies and Applications (2005)
7. Case, J., et al.: Rfc 1157 - simple network management protocol (snmp). Tech. rep.
8. Chen, T.M., et al.: A model and evaluation of distributed network management approaches. IEEE Journal on Selected Areas in Communications 20(4), 850–857 (2002)
9. Chpudhury, R.R., et al.: A distributed mechanism for topology discovery in ad hoc wireless networks using mobile agents, August 11, pp. 145–146 (2000)
10. Inc Cisco Systems: Cisco discovery protocol - cdp (NaN), `http://www.cisco.com/en/US/tech/tk648/tk362/tk100/tsd_technology_support_sub-protocol_home.html`
11. Dudkowski, D., et al.: Architectural principles and elements of in-network management, June 1-5, pp. 529–536 (2009)
12. Dudkowski, D., et al.: Decentralized in-network management for the future internet (2009)

13. Dutta, P., et al.: Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications (2008)
14. Guardalben, L., Mirones, V., Salvador, P., Sargento, S.: A cooperative hide and seek discovery over in network management. In: 2nd IFIP/IEEE International Workshop on Management of the Future Internet (ManFi 2010), IEEE/IFIP Network Operations and Management Symposium (January 2010)
15. Hariharan, S., et al.: Secure neighbor discovery through overhearing in static multihop wireless networks, June 21, pp. 1–6 (2010)
16. Jelasity, M., et al.: T-man: Gossip-based fast overlay topology construction. Computer Networks 53(13), 2321–2339 (2009)
17. Jiang, Y., Lung, C.-H., Goel, N.: A Tree-Based Multiple-Hop Clustering Protocol for Wireless Sensor Networks. In: Zheng, J., Simplot-Ryl, D., Leung, V.C.M. (eds.) ADHOCNETS 2010. LNICST, vol. 49, pp. 371–383. Springer, Heidelberg (2010)
18. Konwar, K.M., et al.: Node discovery in networks. Journal of Parallel and Distributed Computing 69(4), 337–348 (2009)
19. Lee, K.H., et al.: A multicast protocol for network management system. In: Proceedings of IEEE Singapore International Conference on Networks, 1995. Theme: 'Electrotechnology 2000: Communications and Networks'. [in conjunction with the] International Conference on Information Engineering, pp. 364–368 (July 1995)
20. Liu, Y., et al.: A practical hybrid mechanism for peer discovery. In: Proc. International Symposium on Intelligent Signal Processing and Communication Systems ISPACS 2007, November 28-December 1, pp. 706–709 (2007)
21. Moy, J.: OSPF Version 2. IETF RFC 2328 (April 1998)
22. Nassu, B.T., et al.: Topology discovery in dynamic and decentralized networks with mobile agents and swarm intelligence, October 20-24, pp. 685–690 (2007)
23. Inc Overlook: Fing discovery protocol (NaN), http://www.over-look.com/site/index.php
24. Parnes, P., et al.: A framework for management and control of distributed applications using agents and ip-multicast. In: Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 1999, vol. 3, pp. 1445–1452. IEEE (March 1999)
25. Peiwei, C., et al.: The implementation of a new topology discovery algorithm in mobile ad hoc network, October 29-31, pp. 1–4 (2010)
26. Prieto, A.G., et al.: Decentralized in-network management for the future internet. In: International Workshop on the Network of the Future, in conjunction with IEEE ICC 2009, Dresden, Germany (June 2009)
27. Raju, L., et al.: Beacon assisted discovery protocol (bead) for self-organizing hierarchical ad-hoc networks, Novovember 29-December 3, vol. 3, pp. 1676–1680 (2004)
28. Schoenwaelder, J.: Using multicast-snmp to coordinate distributed management agents. In: IEEE International Systems Management Workshop, p. 136 (1996)
29. Sung-Su, K., et al.: Towards management of the future internet, June 1-5, pp. 81–86 (2009)
30. Vasudevan, S.: Self-Organization in Large Scale Wireless Networks. Master's thesis, University of Massachusetts, Dept. Computer Science (September 2006)
31. Wang, J., Li, M., Liu, Y.: Fractured voronoi segments: Topology discovery for wireless sensor networks, November 8-12, pp. 137–145 (2010)
32. Warrier, U., et al.: Rfc 1189 - common management information services and protocols. Tech. rep. (1990)