Lightweight Discovery and Exchange of Network Information in Distributed Network Management

Lucas Guardalben, Tomé Gomes, Rui Sousa, Paulo Salvador and Susana Sargento
Universidade de Aveiro/Instituto de Telecomunicações
Aveiro, Portugal
{guardalben,tomegomes,ruisousa,salvador,susana}@ua.pt

***Abstract*—Node cooperation is a key part of distributed management process, requiring periodic exchange of management information between the nodes. To increase scalability of the network information exchange, network nodes need to cooperate through an efficient set of rules, policies and criteria, to minimize the time and amount of control messages flowing in the network. We propose a solution for optimized decision of the network nodes to perform discovery and exchange of network information, through a Neighbors Eyesight Direction (NED) function and cooperation between the nodes. The proposed solution is embedded into a protocol, named HIde and SeeK Directional Decision (HISK2D), which is compared against current protocols for network discovery and awareness. We show, by means of experimentation, that our HISK2D is more efficient than the base approaches analyzed, in terms of control overhead to discover the nodes, convergence time and exchange of network information according to the acquired knowledge of the network.**

I. Introduction

Emerging networks, as envisioned in Next Generation Networks (NGNs), embrace multiple types of devices and communication technologies, e.g. Ethernet, Wi-Fi, 3G, 4G, etc. Moreover, management of large scale networks is an essential requirement, becoming a challenging situation from the operator's point of view. The operator needs to re-think the management techniques that will be used to respond quickly to this challenge with the purpose of increasing the quality of service for their customers.

This is only achievable through distributed management approaches which bring inherent support for self-management capabilities, enhancing the network with inbuilt intelligence. The current trend of distributed management requires the embedding of management functionalities inside the network nodes. However, this embedding requires network nodes to have sufficient information to take management decisions in an efficient and cooperative way.

Then, this information needs to be exchanged, which may increase the network overhead to unbearable limits. Therefore, new lightweight approaches need to be developed in order to organize the network nodes and determine the interactions between them, to enable optimized network information exchange. In order to deal with the large amount of information exchange, current distributed management solutions use traditional protocols for discovery, communication and cooperation that have difficulties to meet the requirements of low overhead and fast information exchange.

In response to this challenge, we propose the HIde and SeeK Directional Decision (HISK2D) protocol which comprises the several stages of communication, including the discovery of nodes and topology until the exchange of network information. This cooperation is provided by a Neighbors Eyesight Direction (NED) function, that enhances the discovery and exchange information beyond the limit of the directly connected nodes. The benefits provided by this directionality include choosing the best neighbor node to forward the discovery process and exchange of information according to the knowledge depth of the network (e.g. 1, 2 or 3 hops). The role of depth is to ensure the possibility to define the degree of knowledge of a node about its network neighborhood.

We compared the HISK2D protocol with three discovery solutions, namely: Cisco Discovery Protocol (CDP) [1], Optimized Link State Routing (OLSR) [2] and the discovery approach of Open Shortest Path First (OSPF) [3] over an experimentation scenario varying from 4x4 until 9x9 nodes on a grid topology. The results show that the HISK2D is more efficient in terms of control overhead to discover the nodes, convergence time and exchange of network information according to the acquired knowledge of the network (e.g. directed connect nodes or depth larger than 1 hop).

The remainder of the paper is organized as follows. The Section II describes the HISK2D protocol and the implemented modules and their interactions. Section III presents the experimental results. Finally, Section IV concludes the paper.

II. HIde and SeeK Directional Discovery: HISK2D Protocol

The HISK2D protocol was proposed to optimize the discovery process and the amount of information exchanged between the nodes, enabling nodes cooperation in different points of the network. The discovery and exchange of network information in HISK2D are complementary processes: initially, the discovery aims to find the nodes according to the requested knowledge of the network in number of hops (e.g 1, 2, 3...); then, the information gathered through the discovery process is exchanged between the nodes at 1-hop distance, or also through the next hop relay nodes. As part of the nodes discovery mechanism of HISK2D, it was proposed the Hide and Seek concept [4].

The interactions in the Hide and Seek discovery (see Figure 1) are made at 1 hop distance in this initial stage, and they also exchange information about each other, such as identifiers (MAC and Internal control identifier), network status, local resources available (% of free CPU and Memory), source and destination IP addresses. All gathered information is recorded

978-3-901882-50-0 ©2013 IFIP

in a local repository of each *Seeker* node. A node in *Hider* mode (e.g N13) waits for a *Seeker* contact message (or it waits a random time in case it does not receive any message), and then becomes a new *Seeker* cooperating to discover and exchange information with other *Seeker* or *Hider* nodes. This process is repeated until all nodes have been contacted.

However, the information at 1 hop distance may not be sufficient for an efficient distributed management process, since it may benefit from the knowledge of a larger neighborhood. To optimize the amount of information exchanged with low overhead and time at several hops distance, HISK2D uses a Neighbors Eyesight Direction (NED) function, which aims to narrow the directions through which direct nodes or next hop relay nodes will be chosen to continue the discovery and exchange of information.

The NED also brings benefits to distributed management process due to the possibility to control the amount of information to be discovered and exchanged in the network, by enabling the information cooperation between nodes. Therefore, this process depends on relay nodes to forward messages until the depth limit is reached. The role of depth (in number of hops) is to ensure the possibility to define how much a node wants to know about the network, and to decide who is the best neighbor to become the next hop relay node in this depth knowledge. Regarding the difference of the proposed solution with the previous works, this is the one that integrates the NED decision function and the depth, which will provide the partial view (see Equation 2) with different depths.

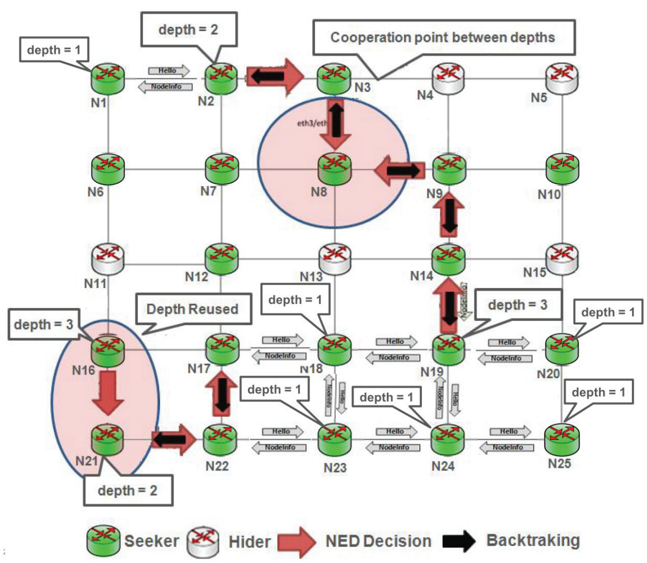

Fig. 1. Example of Hide and Seek and Directional Discovery process using different depths through the Neighbors Eyesight Decision(NED)

The criteria used to choose the next hop neighbor is based on the ranking of the gathered information from the nodes at 1 hop distance. The nodes are also able to maintain the information larger than 1 hop in the repository. The NED decision is based on the link capacity, local node resources available (% of free CPU and Memory) and also nodes density as input parameters, aiming to choose only the 1 hop candidate node that will continue the discovery process, keeping the information gathered between the relay nodes chosen as well. In order to show the NED decision criteria, we define $\mathscr{A}$ as the complete set of nodes in the scenario:

$$\mathscr{A} = \{i : i = 1, \ldots, N\} \tag{1}$$

In any given time instant, all nodes have a local repository named partial view of the network, i.e., the set of nodes discovered from a defined hop distance. The partial view of node i can be defined by:

$$\mathscr{P}_{i,d} = \{j : j \in \mathscr{A} \wedge j \neq i \wedge v(i,j) = d\} \tag{2}$$

where $v(i,j)$ is a function that returns the hop distance (d) between node i and j if there is a path between them, or 0 otherwise.

The link capacity is given by:

$$\mathcal{LC}_i = \sum_{j \in \mathscr{P}_{i,1}} \mathcal{L}_{i,j} \tag{3}$$

where $\mathcal{L}_{i,j}$ represents the reference value of the link capacity $(\mathcal{L})$ from node i to j. This is determined by the interface type and/or software configuration.

Local and network resources are given by:

$$\mathcal{R}_i = \sum_{j \in \mathscr{P}_{i,1}} (\mathcal{C}_j + \mathcal{M}_j) * \mathcal{T}_{i,j} \tag{4}$$

where $\mathcal{C}_i$ represents the available CPU, $\mathcal{M}_i$ the available RAM Memory, and $\mathcal{T}_i$ represents the lowest RTT (Round-Trip-Time) measured by node i communicating to node j.

Network nodes density is given by:

$$\mathcal{N}_i = max|\mathscr{P}_{j,1}|, \forall j \in \mathscr{P}_{i,1} \wedge j \neq i \tag{5}$$

where $max|\mathscr{P}_{j,1}|$ corresponds to the largest Partial View of all connected neighbors, and $|.|$ represents the cardinality of a set.

The global value of the NED is obtained by performing a weighted sum, given by:

$$\mathcal{NED}_i = w_L \mathcal{LC}_i + w_R \mathcal{R}_i + w_N \mathcal{N}_i \tag{6}$$

where, w_L, w_R and w_N represent the weights of (3), (4) and (5). Moreover, the maximum value returned will determine the best neighbor chosen by the node i. Note that each node i will only consider the neighbors $j \in P_{i,1}$ to determine the NED global value (6). Focusing on the factors, it was considered the absolute values for link capacity, RTT and the maximum number of links and interfaces. Additionally, the factors respecting to local node resources, like CPU and RAM memory were normalized, e.g. percentage of CPU and RAM available.

In Figure 1, by default *Seeker* nodes have knowledge at depth 1 hop, being also possible to configure a node with depth larger than 1 hop (e.g N2, N16 and N19). As an example, the initial depth of N19 is 3, and all nodes directly connected to N19 will send their depth and knowledge (partial view) to N19. Then, the NED function will choose a 1-hop neighbor according to the criteria presented in the equation (6) to continue the discovery and exchange of the already obtained information to this chosen node. Then, the best node to continue the discovery is chosen to be N14, and information about N18, N20 and N24 will be exchanged to N14. Node N14 will repeat the 1-hop discovery and will choose a new direction (e.g N9). At this point, node N14 will have information about discovered nodes N9, N18, N20, N24, N19. Then, the next hop chosen is N9, and information about all nodes discovered at this point, plus the ones discovered from N9 (e.g N10, N14) will be sent to the chosen node N8, which reached the depth limit.

All information collected from N19 to N8 will back track to all nodes that belong to the NED paths (e.g N9, N14) up to the node that originated the request (e.g N19). The NED function process ends when: (1) the depth limit previously defined is achieved; or (2) all possible paths between the nodes are explored by the NED function. Note that it is also possible to configure different depth requests in the scenario. For example, consider N2 with depth equal to 2. Then, N2 will send all the gathered information from the nodes N1, N7 to the chosen node N3, and then, node N3 will repeat the process to the next chosen node N8, which reached the depth limit. Node N8 will back track the information gathered up to the node that originated the depth request (e.g N2).

In the case that different NED paths converge to the same node (e.g. N8), this node is considered to be a cooperation point and will back track all information gathered, e.g. from the paths with depths at 2 and 3, to the nodes that originated the requests (N2 and N19). After this process, both nodes (N2 and N19) will synchronize and update the partial views with their directed connected nodes, containing all nodes and its gathered neighborhood hop-by-hop from the NED path direction (e.g N3, N8, N9, N14). After that, all directed nodes already synchronized are able to exchange and cooperate according to a new requested depth.

In addition, in any given time, there may be multiple cooperation points in the network, increasing the efficiency of information cooperation process. Moreover, the NED function can also reuse the knowledge from a node (e.g. N21), without making any further demands. For instance, N16 requests the knowledge with depth equal to 3, and N21 already has this information; then, N21 can reuse the information previously gathered in order to reply to N16. In order to maintain the information always updated, the partial views of each node consider the most recent information exchanged as well.

III. Experimentation Results

A. Virtual Grid Scenario Set-up

The scenario was initially set up as a virtual grid testbed containing OpenWrt Bleeding Edge r28129 (Guest Xen paravirtualized) as virtual machines on a HP Proliant server (CentOS-5 kernel Xen). The testbed is created through a Python script which automatically generates the virtual bridges and link connections between the virtual machines. Experiments are carried out for networks in grid topology with 4x4 (16 machines), 6x6 (36 machines), 8x8 (64 machines) and also 9x9 (81 machines). The results are obtained in two different scenarios: in the first one, the objective is to discover information only from the directly connected nodes; in the second, it is considered the information from neighbors at different depths: 1, 2 or 3 hops. In both scenarios, the interval between periodic packets (e.g. Hello) was configured to be 10 seconds.

In HISK2D, this interval is dynamically adapted with a mean around 10s. The values presented in the graphs below are an average of 5 repetitions and 95% of confidence interval. To perform the packets capture, TCPdump is the software used in each node interface, with an observation time of 10 minutes. The links capacity is considered to be 1 Mb/s. We compare our approach with other well known discovery baselines, using open-source versions of OSPF version 2, CDP version 2 and OLSR version 4. The use of OLSR as one of the baseline protocols is due to the fact that its discovery concept considers complete information of the network (it can also be used in wired networks). This allows to compare the advantages of having the partial or complete information. Moreover, all solutions evaluated share the same events and conditions regarding the programming techniques.

B. Results

The impact of the overhead in a typical operator link is measured and it is shown in Figure 2. Assuming a link capacity of 1 Gbit/s, the percentage of overhead is very low for all protocols. However, the values of the OLSR overhead are very high compared with the other protocols. Figure 2 is presented with a zoom to the other three protocols. As it can be noticed, the impact of the CDP protocol grows proportionally with the network size, and OSPF protocol has a behavior similar to OLSR but with smaller impact.

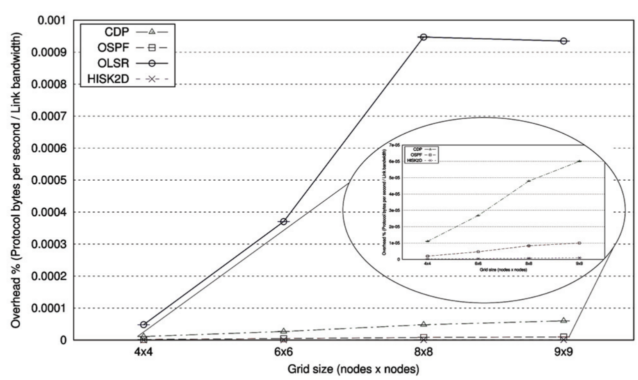

Fig. 2. Overhead using 1Gbit/s link as reference

Regarding the convergence time, OSPF protocol tends to require more time to discover its neighbors, as shown in Figure 3. This was expected, since for large networks, the calculation of Dijkstra's algorithm is complex and consumes significant time. CDP and OLSR protocols have convergence times relatively similar, but comparatively to HISK2D, they all require more time to achieve the same result, i.e., using HISK2D each node discovers more quickly their directly connected neighbors. The chosen signaling mechanism, based on request/response, internal triggers and events, is responsible for this quick performance.

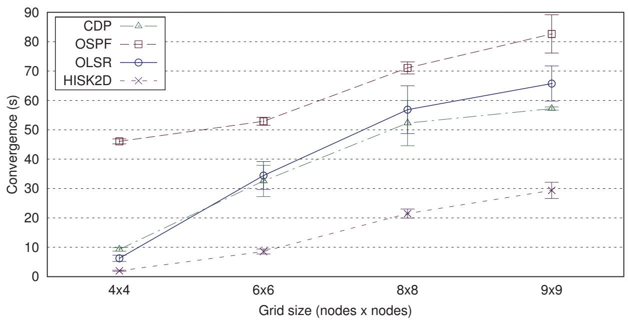

Fig. 3. Discovery convergence time

Figures 4 and 5 analyze the amount of information exchanged and the average convergence time for the second scenario, according to the configured depth. The experiments are carried out in a network with 81 virtual nodes in grid topology. The NED weights are considered to be 33% for each one: link capacity (w1), local node and network resources (w2), and number of interfaces (w3).

This analysis is performed on each machine varying the depth of the discovery process from 1 to 3 hops. In HISK2D, this is directly configured in NED function, which will determine the best interface through which the discovery should proceed. With OSPF, the depth is set using network areas: each node will have its interfaces configured to belong to a certain area depending on the desired depth, i.e., higher values of depth imply larger OSPF network areas.

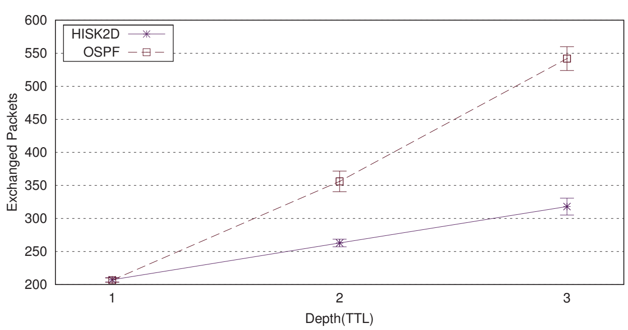

Fig. 4. Grid 9x9: Number of Exchanged Packets

Figure 4 shows that HISK2D requires less packets than OSPF, regardless of the chosen depth. This is mainly justified by the different roles in HISK2D: when the node is *Hider*, it waits to be contacted or starts itself the discovery process after a certain period of time. This is also reflected in the lower amount of data exchanged, although not shown here due to space limitations.

This does not necessarily mean that HISK2D has less information than OSPF, but rather that no redundant data is used, i.e., only the strictly required one is forwarded between nodes. As an example, periodic updates only include the parameters that are changed in the meantime.

To prove that the previously described behavior does not affect the quickness of the HISK2D discovery mechanism, the convergence time is presented in Figure 5. Setting any of the presented depth values, HISK2D is, at least, 60% faster than OSPF.

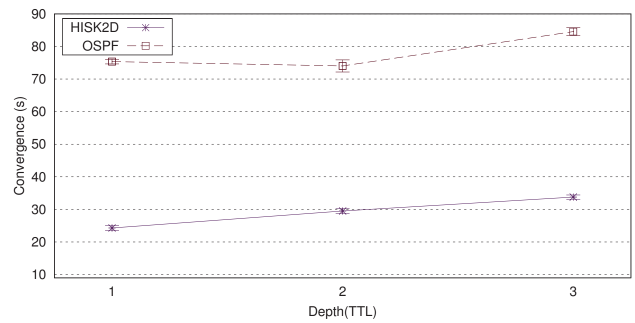

Fig. 5. Grid 9x9: Convergence Time in Seconds to Discovery the Nodes

IV. Conclusions and Future Work

This paper presented HISK2D, a novel protocol for discovery and exchange of network information in large-scale distributed network management. The results present the benefits provided by the HISK2D in terms of control overhead to discover the nodes, convergence time and exchange of network information according to the acquired knowledge of the network. Moreover, the HISK2D proves to be scalable for large-scale scenarios, and also enforces the nodes to search the best neighbor direction through the NED criteria, contributing to improve resource aggregations and reducing the required signaling messages as well. This contributes to a high efficiency in distributed network management. As future work, we plan to adjust the NED weights in real time to optimize the number of discovered nodes or other relevant criteria, according to the network context. We also plan to evolve the implementation and evaluation of HISK2D in wireless scenarios.

V. Acknowledgments

This work was supported by the Fundação para Ciência e Tecnologia - FCT, through the grant SFRH/BD/62511/2009. The authors also thank the support provided by UBIQUIMESH project PTDC/EEA-TEL/105472/2008.

References

[1] inc Cisco Systems, “Cisco discovery protocol - cdp,” 2012. [Online]. Available: http://www.cisco.com/en/US/tech/tk648/tk362/tk100/tsd_technology_support_sub-protocol_home.html

[2] *Optimized Link State Routing Protocol (OLSR)*, RFC 3626, United States, 2003.

[3] J. Moy, *OSPF Version 2*, IETF RFC 2328, April 1998.

[4] L. Guardalben, T. Gomes, A. Pinho, P. Salvador, and S. Sargento, “Nodes discovery in the in-network management communication framework,” in *MONAMI*, 2011, pp. 145–157.