Enabling Inter-Domain Routing in Virtual Coordinate Based Ad Hoc and Sensor Networks

Falko Dressler, Abdalkarim Awad, Reinhard German
Dept. of Computer Science
University of Erlangen
dressler@informatik.uni-erlangen.de

Mario Gerla
Dept. of Computer Science
UCLA
gerla@cs.ucla.edu

ABSTRACT

We discuss challenges arising from using virtual coordinates for routing in ad hoc and sensor networks in multi-domain network scenarios. Recently, it has been shown that virtual coordinate based routing techniques are able to overcome several problems in ad hoc networks. They are independent of known geographic locations and less prone to routing failures. Additionally, some of the proposed solutions are exploiting the concept of Distributed Hash Tables (DHTs) to support data management. A typical example is the Virtual Cord Protocol (VCP). However, it is unclear how to integrate multiple networks or network domains in the concept of virtual address identifiers. We propose a solution for inter-domain routing in virtual coordinate based protocols by exploiting the available DHT. Together with appropriate indirections, efficient inter-domain routing becomes possible.

1. INTRODUCTION

Routing in sensor networks is typically either based on Mobile Ad Hoc Network (MANET) solutions or is using position identifiers of all the nodes. For most scenarios, MANET based routing turned out to be less scalable due to the high resource requirements w.r.t. topology update or flooding based route lookup. Position-based routing solutions inherently improve the situation as simple greedy routing towards the destination can be employed. However, such approaches only work well if the network is dense, as routing holes cause geographic routing to rely on inefficient face routing. Additionally, all the nodes have to be able to precisely obtain their geographic locations.Recently, a number of improvements have been prosed. One idea is to "re-arrange" the nodes' positions appropriately to prevent routing holes. The main idea is to use either location transformation or additional virtual location identifiers. A second concept is to rely on virtual coordinates only. Protocols like Virtual Cord Protocol (VCP) [1] build their own coordinate system that is completely independent of the geographic positions. However, the use of virtual coordinates (that usually follow some well defined constraints) seems to make inter-domain routing extremely complicated.

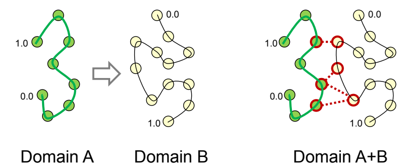

Figure 1: Inter-domain routing between two overlaying virtual coordinate based networks

The problem is illustrated in Figure 1. There might be several possible reasons that two networks may overlay each other. Two different applications in the same geographic area may be kept separated in two virtual networks. However, inter-domain routing might be necessary for efficient communication. Another reason specific to MANETs is group mobility. Two different groups of nodes may operate their own virtual network. If connectivity to other groups can be established, inter-group communication may be achieved by means of inter-domain routing.

The main problem of MANET inter-domain routing has been first discussed in [2]. Four challenging issues have been identified: addressing, membership management, handling domain-level topology changes, and routing between the networks. As Internet-based protocols have been considered, the addressing and membership management basically targeted the IP address assignment procedure and the resulting routing problems. A cluster-based solution for inter-domain routing in MANETs has been described in [3]. Here, especially the issue of domain-level topology changes has been addressed. Using bloom filters, the effort for topology updates was greatly reduced.

Motivated by this work, we investigated this issue for virtual coordinate based routing protocols. In this paper, we show that inter-domain routing in virtual coordinate environments can be established with only marginal overhead. Based on our VCP protocol we outline the basic idea (Section 3) and present some first results demonstrating the feasibility of our approach (Section 4).

2. VIRTUAL CORD PROTOCOL

The Virtual Cord Protocol (VCP) protocol [1] has been developed keeping in mind two objectives. First, efficient routing in sensor networks should be supported and, secondly, data management should be integrated based on the concept of Distributed Hash Tables (DHTs). In previous work, we demonstrated the capabilities of VCP and showed that VCP outperforms MANET based solutions as well as other virtual coordinate protocols [1]. Not yet considered was the case of inter-domain routing.

VCP uses the concepts of DHTs to combine data management with efficient routing in sensor networks. The main idea is to arrange all the nodes in the network in the form of a virtual cord. The topology of this cord must not be "optimal" in any sense because routing is organized by exploiting information about the physical neighbors for greedy forwarding. Nevertheless, the cord ensures a connection between any two nodes. An application-dependent hash function is used for associating data items to nodes. Thus, both pushing to a node and pulling data from a node are supported.

The cord is established using periodic `hello` messages. Beside the assigned virtual address, these messages carry all relevant information including the physical and the virtual neighbors. One node must be pre-programmed as the initial node, i.e. it gets the start position S. Based on received `hello` messages (at least one is required) in the last time interval, a new node can determine its position in the cord. A cord is formed according to a number of simple rules. Basically, new nodes either join at one end of the cord, or get integrated, if at least two other nodes that are virtual neighbors in the cord are detected. A special rule is applied if the node has connectivity to a non-end node but not to its virtual neighbors. Then, a *virtual position* is generated at the discovered potential neighbor that is close to its virtual coordinate. This address allows the new to join between the real and the virtual position in the cord, i.e. to extend the cord without disrupting it.

3. INTER-DOMAIN ROUTING IN VCP

We investigate inter-domain routing in the context of virtual coordinate based routing protocols. This is especially challenging as no fixed intra-domain addressing exists. We target to exploit the available DHT capabilities to solve inter-domain routing using stored gateway identifiers together with indirections. The approach is designed to work with VCP, however, other virtual coordinate based routing schemes can be used as well if they support storing identifiers in an internal DHT.

Conceptually, we associate each network with a unique domain identifier. This can be performed in VCP during the cord setup phase by assigning this ID to the start node. Then, all nodes joining the cord also obtain the domain ID. The periodically exchanged `hello` messages also contain the domain ID. If two networks are getting into each others communication range, a node receiving `hello` messages from another domain automatically becomes a gateway node. It then stores this information into the local DHT by hashing the well-known identifier of the gateway service and storing the information at the node closest to the resulting hash value. If the gateway no longer receives `hello` messages from the detected neighbor, it removes the gateway information from the DHT. This way, the local DHT always contains the most recent gateway information and routing between neighboring domains becomes possible using simple indirections. Whenever a nodes wants to transmit a packet to another domain, it pulls the gateway address from the DHT and then forwards the message to the gateway.

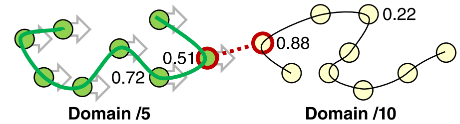

Figure 2: VCP inter-domain routing example

An example is depicted in Figure 2. Two VCP domains are interconnected by a gateway. Let's assume that node 0.72/5 needs to transmit to node 0.22/10. It first looks up the gateway (in our case 0.51/5). It forwards the data to 0.51/5, which, in turn, forwards to the other domain (node 0.88/10) to finally reach 0.22/10.

The routing tables are distributed (and replicated, if necessary) by the DHT. Direct communication between two nodes in arbitrary domains requires global topology information, i.e. the gateway information needs to be distributed into all VCP domains. Inter-domain routing can be supported using a shortest path algorithm together with source routing on domain level. VCP's greedy routing is only used within a domain.

4. SELECTED PERFORMANCE RESULTS

We investigated the feasibility and the performance of the inter-domain routing concept for VCP in a simulation scenario. We used our implementation of VCP for the simulation tool OMNeT++ [1] to analyze the behavior of the dynamic gateway configuration and the performance of the inter-domain routing using indirections.

For a first experiment, we configured two scenarios. The first one consists of 10 stationary nodes and a group of an 10 additional nodes that is moving over the simulation playground. This scenario is depicted in Figure 2. We allow an initial setup time of 50 s to establish two VCP networks, one for each group. Then, the mobile group moves towards the stationary group. After some time, the first nodes get into the radio range of the other group and they start to set up gateway information. After a while, most of the nodes are overlapping until, at the end of the simulation, most of the nodes already

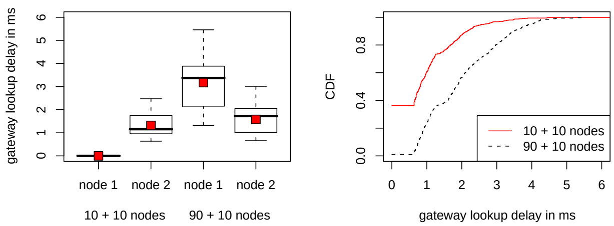

Figure 3: Gateway lookup delay

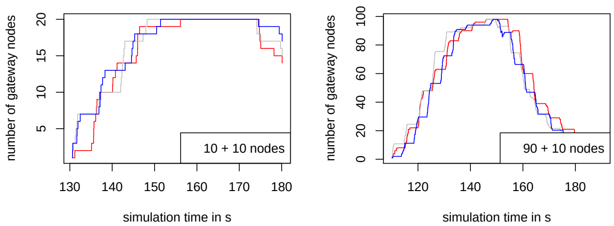

Figure 4: Number of available gateway nodes, plotted is the number of gateways over the time for three independent simulation runs

passed the stationary group. A second scenario used 90 instead of 10 stationary nodes to evaluate the impact of a larger number of gateway nodes and longer communication paths. We performed at least 5 runs for all the simulation experiments.

The first metric that we investigated is the gateway lookup delay. As inter-domain routing is performed using an indirection to identify a gateway to the neighboring VCP cord, this gateway ID needs to be looked up in the local DHT. Figure 3 shows the simulation results. In the boxplot (which shows the median as a thick line and a box indicating the 1st and 3rd quartiles, as well as the mean value depicted as a small red marker), the lookup delay for two nodes in both scenarios is drawn. Obviously the lookup delay varies depending on the logical position of the node in the cord. The observed delays in the smaller group (10 nodes) is obviously smaller compared to the larger group (90 nodes). This trend is confirmed by the plotted CDF.

Furthermore, the number of available gateways is of interest as this finally can be regarded as a measure of the inter-domain routing complexity (the more gateways, the more difficult is the selection of a "best" gateway). Figure 4 shows the results for three runs. As both groups of nodes approach each other, new gateways are added. The maximum number (20 or 100 gateways, respectively) is reached for a small time interval. Then, both networks are departing again. Finally, we measured the overall communication delay for inter-domain routing as depicted in Figure 5. Here, the communication between the source node and the gateway as well as the additional delay from the gateway to the destination node

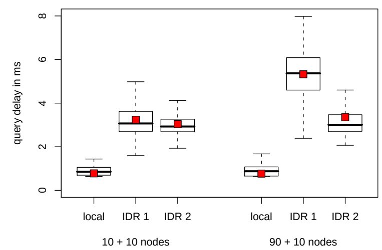

Figure 5: Query delay (local and inter-domain)

is measured. The figure compares local communication within a single cord with inter-domain routing (depicted are results for two selected nodes).

5. FURTHER CHALLENGING ISSUES

Global topology maintenance can be supported by periodic updates of the routing tables in all domains. However, this only works for very small numbers of domains. For larger networks, either a bloom filter based approach similar to [3] may be used or even a global DHT.

Identification of different VCP domains is currently based on pre-defined unique IDs. In very dynamic scenarios, even random identifiers may be helpful. For example, domain partitioning and merging are difficult to integrate using static IDs. One solution is to start with initially assigned unique IDs as described in [3]. Domains may split into partitions and then use the original ID as a prefix. Merging is then only supported for the original groups.

Split detection can be supported using the periodic `hello`'s for neighborhood management. Domains split if no further `hello`'s are received after some time threshold. The main problem is to distinguish between a node failure and a domain split. *Merging* on the other hand is simple if two domains belonging to the original ID meet again. However, the stability of the cord needs to be ensured and in most cases (e.g., merging in the middle of the cord), a renumbering might be needed.

6. REFERENCES

[1] A. Awad, L. R. Shi, R. German, and F. Dressler. Advantages of Virtual Addressing for Efficient and Failure Tolerant Routing in Sensor Networks. In *IEEE/IFIP WONS 2009*, pages 111–118, Snowbird, UT, February 2009.

[2] C.-K. Chau, J. Crowcroft, K.-W. Lee, and S. H. Y. Wong. Inter-Domain Routing Protocol for Mobile Ad Hoc Networks. In *ACM MobiArch 2008*, pages 61–66, Seattle, WA, 2008.

[3] B. Zhou, Z. Cao, and M. Gerla. Cluster-based Inter-domain Routing (CIDR) Protocol for MANETs. In *IEEE/IFIP WONS 2009*, pages 19–26, Snowbird, UT, February 2009.