Efficient Service Discovery in Sensor Networks using VCP

Abdalkarim Awad, Reinhard German and Falko Dressler
Computer Networks and Communication Systems
Dept. of Computer Science, University of Erlangen, Germany
{abdalkarim.awad,german,dressler}@informatik.uni-erlangen.de

ABSTRACT

We show the feasibility of efficient routing and service discovery in sensor networks using the Virtual Cord Protocol (VCP). Scalable and energy efficient data management is still a challenging topic in this domain. Recently, the advantages of virtual coordinates have been explored in comparison to solutions based on geographical positions. Due to the fact that such virtual coordinates can be assigned in an "optimal" way, routing and data management becomes more efficient. We implemented our VCP protocol on BTnode sensor nodes to show the applicability of this protocol in a lab environment. Furthermore, we address the issue of service discovery by means of indirections as known from overlay networks.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) represent a class of wireless networks that strongly relies (among others) on energy efficient operation and self-organizing management and operation. These aspects are especially challenging in the field of dynamic service discovery. Basically, these have to be addressed within multiple layers, or more precisely, through cooperation between these layers. Most relevant are *routing techniques* and the corresponding node identification or *service discovery procedures*. The scenario is depicted in Figure 1. In a first step, the appropriate node needs to be identified. Then, routing techniques are required to find an adequate path to the service location.

Routing approaches for WSNs are still dominated by table-driven solutions known from Mobile Ad Hoc Networks (MANETs), even though these might be not scalable due to the high costs for topology discovery and network management issues. Furthermore, service discovery is mainly based on either pre-programmed addresses or publish-subscribe techniques using centralized service brokers.

Both problem domains are being addressed by recent Distributed Hash Table (DHT) based approaches for data management in WSNs. In general, such solutions can be classified in three main categories: real location based, virtual location based, and location independent.

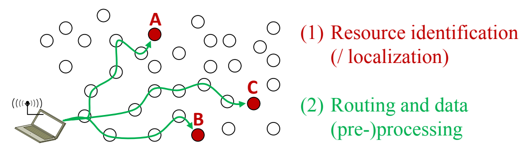

Figure 1: Problem description: service identification and efficient routing

Geographic routing is conceptually an excellent idea for use in sensor networks. The required geographic coordinates can be gained either through GPS or by means of self-localization. Geographic Hash Tables (GHTs) [4] hash keys into geographic locations, so the data items are stored on the sensor node geographically nearest the hash of its key. For routing, protocols such as Greedy Perimeter Stateless Routing (GPSR) are used that use the physical location of nodes. Unfortunately, geographic routing does not ensure adequate results in all topologies. Problems with dead ends and costly recovery procedures make these approaches suitable for selected application scenarios only.

Most recent approaches for data management and routing in WSNs rely on virtual coordinates. Inspired by DHTs and bringing this idea down to the underlay, efficient routing paths can be maintained together with the capability to store information and data in a DHT-like system. The first solution in this field has been Virtual Ring Routing (VRR) [2]. It uses a location independent unique key to identify nodes and to organize them into a virtual ring. For routing purposes, each node maintains a set of virtual neighbors of cardinality r that are nearest to node identifier in the virtual ring. For packet forwarding, VRR picks the node with the identifier closest to the destination from the routing table and forwards the message towards that node. The problem of this protocol is that the adjacent nodes in the ring can be far away in the real network. Moreover, dead ends cannot be completely prevented.

Recently, Virtual Cord Protocol (VCP) has been proposed, which shows a couple of advantages compared to VRR [1]. The concept of the protocol always ensures reachability of all destinations and optimizes routes on the fly by exploiting information about physical neighbors. Using the concept of indirections, VCP also provides means for efficient resource and service discovery. In particular, VCP exploits the inherent use of a DHT to organize data in the network together with indirections to store and to retrieve service locations in a publish/subscribe manner. Node providing services publish this information in the virtual cord by means of hashing the information to a particular destination and storing the node's location at that destination. Then, nodes may use (subscribe to) the service by hashing the service again, retrieving the service's location (i.e., following the indirection), and finally accessing the service. A similar approach has been described by Jung et al. [3]. However, a separate overlay is used in this solution that requires additional effort for underlay routing.

This demo shows the applicability of VCP for service discovery and efficient routing in sensor networks. After evaluating the protocol using comprehensive simulation studies in previous work [1], we implemented the protocol on the BTnode sensor node platform, which is based on an Atmel ATmega128 micro controller and a CC1000 radio transmitter. We show the feasibility of the protocol implementation in a small but easy to understand demo scenario.

2. SERVICE DISCOVERY WITH VCP

The Virtual Cord Protocol (VCP) exploits 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 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 the push and the pull principles are supported – pushing to a node and pulling data from a node. The same concept can be used for reliable service discovery in massively distributed systems.

2.1 Cord setup and routing

In the following, we briefly introduce the cord setup and routing process. More details and the complete algorithms can be found in [1]. One node must be preprogrammed as initial node, i.e. it gets the start position S. In the current version of VCP, `hello` messages are used to discover the network topology. Besides the assigned virtual address, these messages carry all relevant information including the physical and the virtual neighbors. Based on all received `hello` messages (at least one is required) in the last time interval, a new node can determine its position in the cord.

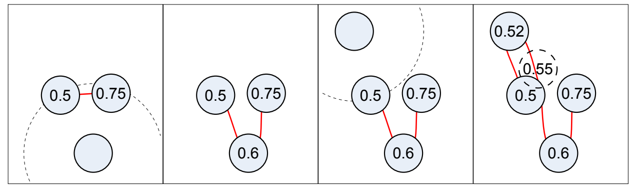

Figure 2: Basic join operation for an intermediate and a "virtual" node

If the new node can communicate with an end of the cord, it just becomes successor or predecessor of this node. If at least two other nodes that are virtual neighbors in the cord are detected, the new node gets a virtual position in between these two. If this is not the case, i.e. the new node has contact to at least one non-end node and to no virtual neighbors in the cord, then the new node asks its neighbor node to create a co-located virtual position. The new node then gets a position between the real and the virtual position of its neighbor. Figure 2 outlines the join process for a normal join (steps 1+2) and for the use of a virtual position (steps 3+4).

2.2 Service discovery

Service discovery and data management are handled by the protocol inherent capabilities to identify virtual coordinates of "items" in the network. An application-dependent hash function is used to hash specific data to nodes in the network. This functionality supports the classical push and pull principles to access data in the sensor network, i.e. the hash value of an identifier can be used to store data at this position in the cord or to retrieve data, respectively.

Similarly, service discovery can be organized by hashing a service description to a virtual position. At this position, a link can be stored that points to the address of the node that provides this service. Thus, using indirections like in overlay networks, VCP can identify and locate services in the sensor network as well as route

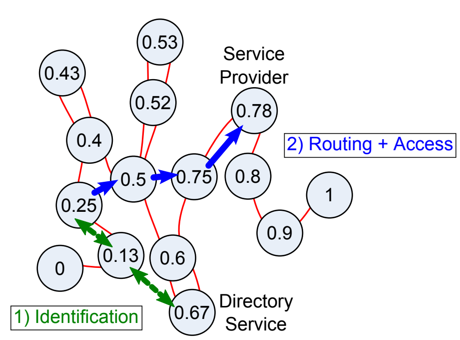

Figure 3: Service discovery using indirections and VCP routing

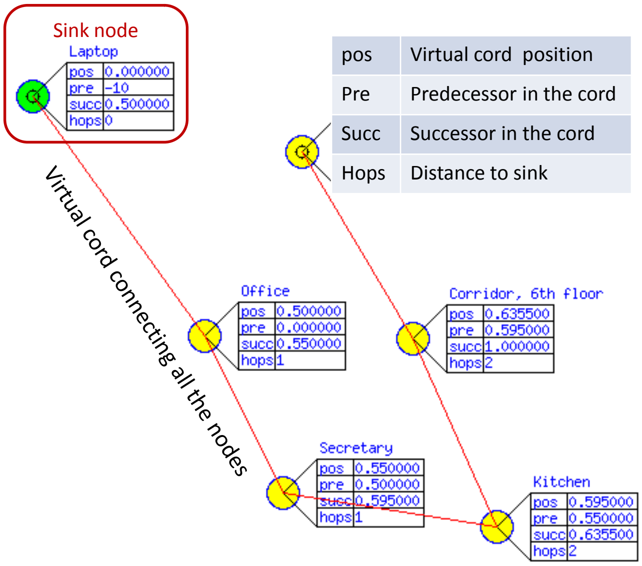

Figure 4: Cord visualization in the lab demo

packets to the identified service provider. Specifically, nodes may store their virtual coordinate, that is the local VCP address, at H(service). Basically, this concept follows the publish/subscribe paradigm. Services announce their presence by publishing their identity in the DHT. Then, nodes my use (subscribe to) the service by retrieving the service's virtual coordinate and using VCP to route to the service.

The concept is shown in Figure 3. VCP connects all the nodes in a virtual cord. This cord is specified in the setup phase by allocating virtual addresses to each node. These addresses in the range $[0, 1]$ are used together with a hash function mapping items to virtual node addresses. In our example, 15 nodes are depicted. In a first step, the service provided by node 0.78 is stored in the DHT (in this example, the service ID is 0.67). Access to the service is then requested by node 0.25. It first uses the same hash function to find the directory service for the particular service. So, it first places a request to node 0.67 to discover the cord address of the service provider. Then, it is able to directly access the service provider (node 0.78) to access the service.

3. IMPLEMENTATION AND DEMO

One of the main questions about VCP was concerning with the feasibility of an implementation in a real world scenario. Therefore, we started an implementation project with two intentions. First, in order to prove the feasibility of VCP to operate on low resource sensor nodes and, secondly, to identify possible issues due to hardware limitations.

For the implementation, we chose the BTnode sensor platform. It is comparable to the widely used Mica mote series. An Atmel ATmega 128 micro controller builds the basis of this node supported by a CC1000 radio transceiver and a Bluetooth module. We used the BTut operating system, which has been developed based on the open source Nut/OS. Similar to TinyOS, it is a real-time embedded operating system compatible with ATmega processor family. In addition to minimal OS services, it provides a multi threaded system architecture. We used this thread support to schedule tasks for **hello**, **insert**, **query** messages. The used BTnode sensor nodes primarily use the Berkeley MAC (BMAC) protocol.

The size of the compiled system to be installed on the BTnode is about 87 kByte. The file contains both the text and the data sections. The text portion contains the actual instructions, while the data contains the program's data part. The resulting file size is reasonable for sensor nodes of this class. To simplify the administration of the network we implemented some functions and registered them to be called from the terminal:

- `printN` displays some debugging information information about the node (like its virtual position),
- `setpos 0` is used to initialize the first node, and
- `sink 1` assigns the current node as a sink.

In our demo, we are going to show the setup of the virtual cord. We connect a single node to a laptop that announces a new service to the network: being a sink for network management information. All other nodes query the hash table for the current position of the sink node and, if successful, start sending their routing table entries to the sink. Using this information, the laptop graphically shows the connected nodes, their cord addresses, and the cord structure. The setup is shown in Figure 4.

4. REFERENCES

[1] A. Awad, C. Sommer, R. German, and F. Dressler. Virtual Cord Protocol (VCP): A Flexible DHT-like Routing Service for Sensor Networks. In *IEEE MASS 2008*, pages 133–142, Atlanta, GA, September 2008. IEEE.

[2] M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron. Virtual Ring Routing: Network routing inspired by DHTs. In *ACM SIGCOMM 2006*, Pisa, Italy, September 2006.

[3] J. Jung, S. Lee, N. Kim, and H. Yoon. Efficient service discovery mechanism for wireless sensor networks. *Elsevier Computer Communications*, 31(14):3292–3298, September 2008.

[4] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-Centric Storage in Sensornets with GHT, a Geographic Hash Table. *ACM/Springer Mobile Networks and Applications (MONET), Special Issue on Wireless Sensor Networks*, 8(4):427–442, August 2003.