Dynamic Address Allocation for Management and Control in Wireless Sensor Networks

Zheng Yao and Falko Dressler
Autonomic Networking Group, Dept. of Computer Science 7
University of Erlangen-Nuremberg, Germany
dressler@informatik.uni-erlangen.de

Abstract

Several data-centric communication paradigms have been proposed in the domain of wireless sensor networks (WSN). Therefore, the principles of operation and maintenance in such networks are changing in order to control massively distributed systems. Previous addressing schemes fail or produce too much overhead even if only locally unique addresses of sensor nodes are required. In this paper, we present a dynamic address allocation scheme for localized address assignments in WSN. We developed a round-based address assignment with subsequent duplicate address detection that operates in a self-organized manner. It inherently allows busy-sleep periods and does not assumes always awake nodes. In order to verify the approach, we implemented the algorithm on Mica2 sensor motes and tested it in a WSN maintenance scenario. The results demonstrate that our method works well for operation and maintenance of WSN without prior address assignments.

1. Introduction

In this paper, we propose a novel scheme for dynamic address allocation in wireless sensor networks (WSN). This allocation scheme was specifically developed for maintenance operations in large-scale networks, i.e. it works on a local set of nodes instead of building network-wide unique address tables. Therefore, it is more scalable and efficient than competitive techniques.

Ad hoc communication has become a major subject in the networking community. Especially, WSN have become a research target due to their specific requirements and restrictions [1]. Research challenges and further directions include energy-aware operation, scalability, and optimized resource management [5]. Besides physical resources such as memory, processing power, and energy, logical resources in terms of naming, addressing, and topology control must be organized and controlled.

Self-organization is regarded to be the new paradigm for operation and control in ad hoc networks and WSN [6]. Without the need of global state maintenance, the major objective *scalability* can be easily addressed. While methodologies for generic self-organization are still future vision, communication and routing aspects are well-covered so far.

Usually, current network layer protocols require an unique addressing for all nodes in the network. Various routing protocols have been proposed. Nevertheless, in ad hoc and sensor networks, the data communication, routing strategies, and topology management is very scenario dependent, i.e. different solutions are optimal for, e.g. agriculture [2] and habitat monitoring [16] scenarios. A survey of ad hoc routing protocols can be found in [4]. The overall objective for all these solutions is to develop scalable routing protocols for application in WSN scenarios consisting of a huge number of communicating nodes [11, 12].

Interestingly, many scenarios in the domain of wireless sensor networks do not depend on fixed node addresses due to various reasons:

- The application itself does not require globally unique addresses. For example, measurement results need to be transmitted and analyzed in an sensor/actuator network. The application only need to know about the region or position of an identified event but the address of a particular node.
- The deployment and maintenance becomes much easier if address-less or data-centric operations are enabled. Nodes can be deployed (or replaced) without changing the node program.
- Even if dynamic addressing on a global base is considered, the overhead due to the address assignment process, either during development or replacement can be too high. Such overhead is caused by address assignment protocols to find network-wide unique addresses.

1530-1605/07 $20.00 © 2007 IEEE

These finding lead to many proposals for data-centric routing approaches. Basically, most of them are based on flooding schemes [14, 15, 17]. Probabilistic solutions using the idea of stochastically reducing the overhead caused by flooding approaches are very successful. The most prominent solution is gossiping [3, 10]. Similarly, gossiping is used for resource location protocols [13]. Finally, geographical routing is an example of address-less operation even if this is not an example for data-centric routing [24].

The data-centric operation principles allow an efficient data communication. Nevertheless, during *operation and control*, specific nodes need to be addressed in order to update software modules, to calibrate sensors, to perform localization tasks, and others.

In this paper, we propose and evaluate an algorithm for localized address allocation and management. This scheme is a major building block in an overall operation and maintenance scenario. The scheme is round-based and inherently allows busy-sleep periods and does not assumes always awake nodes. We also implemented the method in a testbed consisting of Mica2 sensor motes and several mobile robot systems maintaining the sensor network. These robots use the address allocation scheme to identify specific nodes that need to be calibrated or re-programmed.

The rest of the paper is organized as follows. Related work is discussed in section 2. The algorithm and operation details are presented in section 3. Then, we discuss an application scenario in section 4. Section 5 concludes the paper.

2. Related Work

Solutions for dynamic address allocation have been proposed in various contexts. The best known example is DHCP (Dynamic Host Configuration Protocol) for IP and IPv6 networks, respectively [7, 8]. Similarly, techniques for operation in much more dynamic environments such as mobile ad hoc networks have been proposed. A detailed summary can be found in a comprehensive study of dynamic addressing schemes [18] and a paper providing an overview and future directions for such address allocation solutions [22]. In the context of ad hoc and sensor networks, PACMAN (Passive Autoconfiguration for Mobile Ad hoc Networks) needs to be mentioned as an optimized solution for efficient dynamic address allocation [21]. This approach is directly based on the lessons learned from the DHCP development.

In the following, we shortly discuss two solutions that we used as a starting point for developing our novel localized address allocation scheme: DHCP and PDAD.

2.1. Dynamic Host Configuration Protocol

DHCP [7, 8] is a client-server based network protocol. It consists of two major building blocks: a protocol for delivering specific parameters to the client and a mechanism for selection and suggesting IP addresses. Each DHCP server maintains one or more address pools. Such a pool describes available and currently used addresses, respectively. The address management is server-oriented. This working principle ensures the uniqueness of assigned IP addresses. In consequence, no detection scheme for duplicate addresses is necessary.

DHCP supports three different mechanisms for IP address allocation: automated, operator-controlled, and dynamic. In the first case, an IP address is permanently assigned to a client after its first registration. Similarly, an operator can manually assign an address to a particular client. Finally, dynamic address allocation provides the possibility to temporally bind an IP address to a client. This assignment is limited in time and must be renewed after a given lease time.

The dynamic assignment is the only possibility to reuse addresses after a first allocation. Thus, dynamic address allocation is usually used if only short-term connections of clients are envisioned. Focusing on the management and control in ad hoc and sensor networks, this seems to be an adequate solution. Unfortunately, all the assignments are based on the MAC address of each client that is worldwide unique. Therefore, DHCP only maintains an IP address to MAC address binding.

2.2. Passive Duplicate Address Detection

PDAD [20] was specifically developed for mobile ad hoc networks. Basically, Passive Duplicate Address Detection is not an algorithm for choosing an address but for detecting duplicates. The primary objective during the development of PDAD was its application in networks with high node mobility. Therefore, the result was a light-weight scheme for address allocation with passive duplicate detection.

The behavior of the algorithm is as follows: each node randomly choses an address by itself. There is no need for a particular server or any other required network infrastructure. In a second step, the node performs the duplicate address detection (DAD) algorithm to verify the uniqueness of the selected address by passively observing the network traffic. PDAD continuously checks routing information for bandwidth-efficient DAD, e.g. sequence numbers are verified.

In summary, PDAD provides an efficient address allocation algorithm that depends on particular ad hoc routing information. In case of data-centric data communication, additional messages must be created to enable PDAD in

such networks. Obviously, this results in unnecessarily high overhead. Additionally, PDAD's objective is to maintain globally unique addresses. In many scenarios, there is no such requirement.

3. Algorithm and Methodology

The proposed algorithm for dynamic address allocation benefits from many ideas learned from DHCP and PDAD. A selected device, e.g. a server performing management and control operations, initiates and controls the address assignment process. This server maintains a list of previously allocated addresses. While this behavior looks similar to DHCP, there is no binding to any kind of hardware address or other unique identification of a participating sensor node. Therefore, an extension, duplicate address detection (DAD), is necessary. After such an address allocation step, the server can continue in maintaining the surrounding sensor nodes by contacting each of them individually.

The envisioned scenario allows multiple server nodes performing management and control actions concurrently. Additionally, we assume mobility and spontaneously emerging or failing nodes. Therefore, the address assignment cannot last for an unlimited period of time. Nevertheless, we do not enforce periodic re-assignments in order to prevent the disruption of running maintenance operations such as node reprogramming. Instead, a round-based system is used that creates a logical ordering of time information in a local environment. If a new round is started, all nodes must update their addresses while sub-rounds are used to discover new nodes that appeared in the surroundings.

Initially, the server cannot communicate with particular nodes using unicast communications because the server cannot assume a previous address-assignment step. Therefore, broadcast messages are employed to discover neighboring nodes and to initiate the address allocation. In ad hoc and sensor networks, such a broadcast is also the most efficient way to reach all surrounding nodes with a single radio packet.

After the first allocation step, the server can seamlessly continue with its management and control operation using unicast communication targeting specific nodes. For example, nodes might be calibrated, reconfigured, or reprogrammed. In order to prevent disruptions, measures have to be taken in order to ensure the uniqueness of the selected addresses in the local environment. This also includes the possibility for spontaneously arriving nodes and even busy-sleep cycles without demanding always awake nodes.

In the following, we depict the involved processes during the address selection, duplicate prevention, and address management. Exemplary, we implemented the scheme on Mica2 sensor motes running TinyOS (see also section 4). In the wireless sensor network community, this system is a quasi standard in the academic world.

3.1. Generation of random addresses

Each sensor node must be equipped with a small module for the execution of the dynamic address allocation procedure. This module implements an interface that waits for messages that initiate the address assignment. The complete communication protocol is shown in section 3.6. After receiving a message with a new round id, the node randomly chooses an address and assigns it to its radio interface. In TinyOS, component `RandomFSR` is used that provides the possibility to generate a random number in the range $[0, 2^{16}]$. This address is sent to the server that initiated the address assignment for verification, i.e. the check for duplicate address allocations. Therefore, the implementation of the node itself is very lightweight (a packet handler and a random number generator).

3.2. Dynamic Address Allocation

The dynamic address allocation is always started by a dedicated node. In the following, we call this node *server* because we assume that this node will provide management and control operations for the neighboring sensor nodes (which is the only reason for maintaining unique addresses). The complete allocation procedure is depicted in figure 1 and explained in the following.

First, the server broadcasts a message with the current round id. All neighboring nodes receive this broadcast in order to check the round id to verify whether they are involved in this address allocation round or not. Then, they choose an address and transmit it to the server using a HELLO message.

In this context, the round id is an identifier for a set of activities. Identical addresses that were registered in different rounds have disjunct meanings. Therefore, a sensor node must discard a message containing the same address that its own but carries a different round id. Only messages with the same round id can be accepted and processed.

There are different possible problems that can appear in such a round leading to nodes without a proper address assignment. First, messages can get lost. This includes the first message from the server that initialized the new round as well as for the response from a sensor node that informs the server about its temporary chosen address. Secondly, multiple nodes might choose the same address and this duplicate is detected by the server. All these cases require an additional assignment step.

We solve these issues by using the following procedure: after a round is started, the server can spontaneously reinitiate the dynamic address allocation procedure in order to

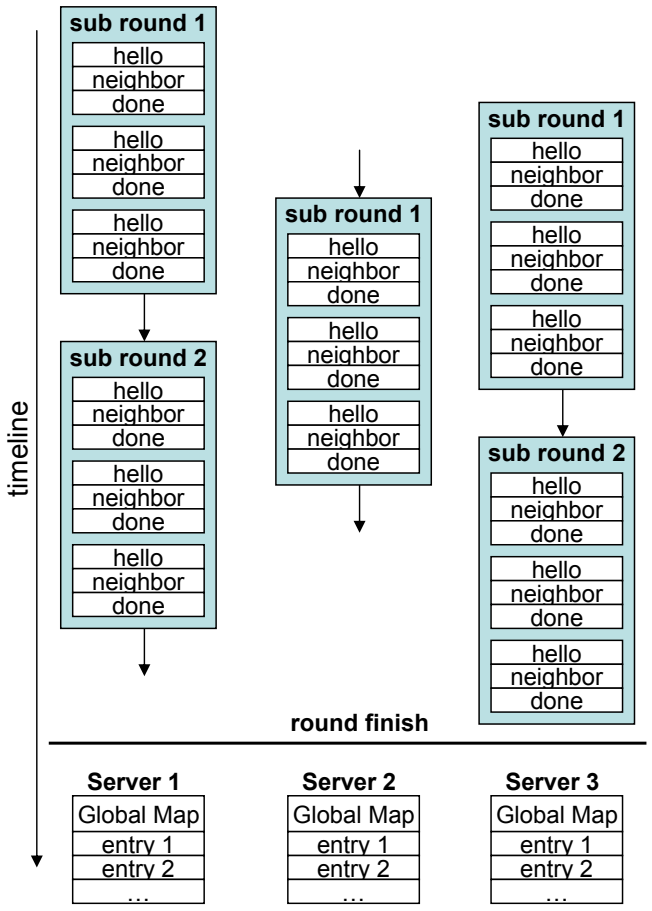

Figure 1. Round based address selection

assign addresses to nodes that have not yet been processed. We call this step a *sub round*. As shown in figure 1, each sub round allows the replicated execution of the dynamic allocation procedure. This behavior was implemented in order to prevent duplicate address assignments.

In each sub round, a sensor node responds to a HELLO message with a randomly chosen address. This address is verified by the server whether it is already registered or not. If the address is known, the server enforces the sensor node to choose a new randomly generated address by sending a REDO message. If the message is unique, the server finally assigns the address to this particular node by sending an ASSIGN message. If an ASSIGN is received by a sensor node, this address is finally allocated and registered by both, the node and the server. On the other hand, a REDO re-initializes the process. All nodes that already allocated an address in this round must ignore and discard further messages for the dynamic address allocation process.

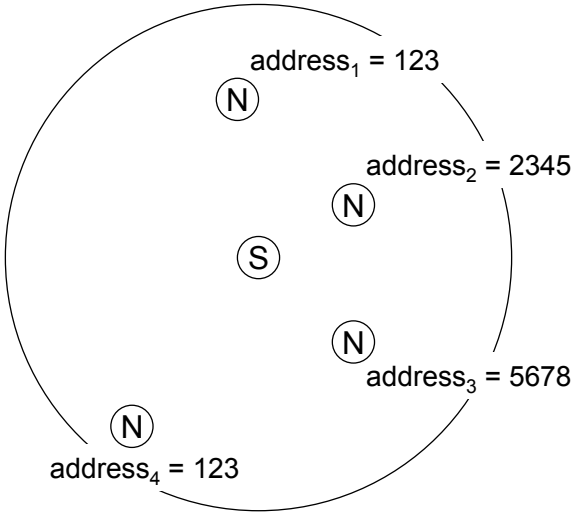

Figure 2. Possible duplicated in a sub round (S - Server, N)

3.3. Possibilities for Duplicate Addresses

We discovered three possible cases how to produce duplicate addresses:

1. *Duplicate addresses in a single sub round:* Such duplicates can appear if two nodes spontaneously choose the same address after the reception of initiation message from the server. Both nodes respond at the same time to the server. Therefore, the server detects the duplicates.

   Figure 2 depicts this case. Two nodes randomly choose the address 123, i.e. $address_1$ and $address_4$ have the same value. This collision can be corrected either within the same sub round or in a second sub round. Again, there is some probability that the same two nodes choose the same address. Nevertheless, the probability is very low (see below) and decreases with each sub round.

   Assuming n surrounding nodes and a available addresses, the probability of an address collision is

$$P_{duplicate} \sim 1 - \frac{a!/(a-n)!}{a^n} \qquad (1)$$

Proof
Because there is no synchronization between the nodes, more than one node can assign the same address in each sub round. Therefore, the number of permutations without one or more duplicates is $P_a^n = \binom{a}{n} * n! = \frac{a!}{n!(a-n)!}$. Thus, the probability to

find a combination of addresses without duplicates is $P_{noduplicate} = P_a^n/a^n$. This leads to the equation for the probability of having duplicate addresses $P_{duplicate} = 1 - P_{noduplicate}$ as shown in equation 1.

For validation of the algorithm, we assume a server with n surrounding nodes and an address length of 16 bit (`uint32_t`), i.e. a total number $a = 2^{16} = 65536$ possible addresses. Then, the probability of duplicates in a round will be about 0.00068 for 10 nodes, 0.0028 for 20 nodes, and 0.072 for 100 nodes (which are usual numbers in sensor networks with an address length of only 16 bit).

The obvious conclusion is the more sensor nodes are in the surrounding, i.e. the radio range of the server, the higher is the probability to find duplicate addresses in a sub round. Such duplicates cannot be prevented. Nevertheless, the probability can be controlled by verifying the application scenario and adapting the address range according to it. Additionally, the method for choosing the address can be changed from a random process to one that maintains history information. Such a process would decrease the probability of having duplicate address allocations in a sub round (not preventing it). Nevertheless, depending on the algorithm, there will be a remarkable overhead at the sensor side.

2. *Duplicates in different sub rounds:* If a sensor node chooses an address in a sub round that was already allocated by another node in a previous sub round, the server detects this duplicate because it is already registered in its GlobalMap of already-assigned addresses. Therefore, the server responds with a REDO message that leads to a new sub round for at least all sensor nodes that randomly chose this particular address, i.e. these nodes do not get an assigned address in this sub round and perform the overall operation once again.

3. *Duplicates in NodeMaps from different Servers:* If multiple servers concurrently initiate the address allocation process and both select the same round id, it may happen that different sensor nodes allocate the same address controlled by different servers. All these servers register this particular address in their GlobalMap. Therefore, duplicates in different GlobalMaps may appear (even if the probability is quite low). Counter measures must be applied to encounter such problems.

3.4. Prevention of Address Duplicates

In the previous section, we discussed three possible cases how to generate duplicate addresses. In this section, we briefly describe solutions to prevent or at least to detect and react on address duplicates.

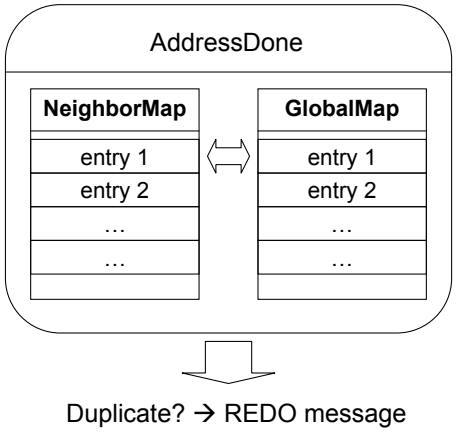

Figure 3. Local duplicate detection and according response (REDO message)

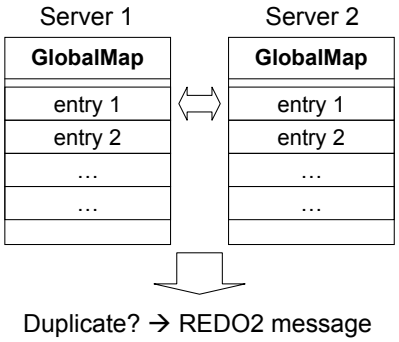

Figure 4. Distributed duplicate detection between multiple servers (REDO2 message)

To enforce duplicate addresses caused by case (1) or (2), i.e. duplicates in a single or in subsequent sub rounds, the server compares selected addresses as stored in a local NeighborMap with already allocated addresses as stored in the GlobalMap. This procedure is depicted in figure 3. A REDO message is sent in case of duplicate detection, i.e. a new random number will be generated by the corresponding sensor nodes and proposed for selection in the following sub round.

Counteracting duplicated cased by case (3), i.e. multiple servers concurrently assigning addresses to nodes in the same environment is more difficult. As shown in figure 4, a distributed duplicate address detection must be initiated. We assume that such servers will have resources available

(storage, CPU, energy) much larger compared to the sensor nodes. Therefore, such an address verification step can be done easily with well-established data synchronization schemes. If a duplicate is detected, a special message, REDO2, is sent to the sensor nodes in order to re-assign addresses even if they were already committed.

3.5. Message Types

Every data packet used for the communication between the server and the sensor nodes is identified by an id. Based on the event-driven programming paradigm of typical sensor APIs, such an id is mapped to an active message handler id. The actions of the dynamic address allocation scheme as shown in the next section are triggered and controlled by these messages. The following message ids are defined and used by our address allocation algorithm:

- `DYNAMICADDRESS_TYPE_ROUNDID` – A new round is initiated with this message. Previous assignments must be discarded by every node.
- `DYNAMICADDRESS_TYPE_HELLO` – The HELLO message is used to discover all nodes in the direct neighborhood of the server. All nodes randomly choose an address and send this address to the local server in this message.
- `DYNAMICADDRESS_TYPE_VERIFY` – This address verification message contains the randomly selected address of the node for later permanent assignment by the local server.
- `DYNAMICADDRESS_TYPE_REDO` – An address duplicate was detected in a sub round or in subsequent sub rounds. The nodes without a previously assigned address that are receiving this message must perform the address selection step again.
- `DYNAMICADDRESS_TYPE_ASSIGN` – Selected addresses are acknowledged with the ASSIGN message. After the reception of this message, the node finalized the address assignment step and allocates the selected address to its radio interface.
- `DYNAMICADDRESS_TYPE_REDO2` – An address duplicate was detected between multiple servers. The REDO2 message informs all surrounding nodes that they must discard previously assigned addresses and initiate the allocation scheme again.
- `DYNAMICADDRESS_TYPE_RETURN` – For verification and debugging issues, this message can be used to request all neighboring nodes to submit their currently assigned addresses.

3.6. Execution Scheme

The execution scheme is depicted in figure 5. In the following, all shown steps are explained:

Step 1 (AddressHello): A message of type `DYNAMICADDRESS_TYPE_ROUNDID` is used to initiate a new round. All nodes receiving this message discard their previously assigned messages and get ready for the HELLO exchange. After the reception of the message of type `DYNAMICADDRESS_TYPE_HELLO`, each sensor randomly chooses an address and sends it to the local server in a `DYNAMICADDRESS_TYPE_VERIFY` message.

Step 2 (AddressNeighbor): The server collects all incoming `DYNAMICADDRESS_TYPE_VERIFY` messages and stores this information in a temporary map, the NeighborMap. This map is later used for duplicate address detection.

Step 3 (AddressDone): In this step, the server searches for duplicate addresses in its NeighborMap. If all selected addresses in the last sub round are unique, the server copies these entries into its GlobalMap containing all allocated addresses. Additionally, it sends messages of type `DYNAMICADDRESS_TYPE_ASSIGN` to the associated nodes. These nodes finally assign the selected addresses to their radio interface. In Mica2 motes, this is done by binding the selected address to the TOS_LOCAL_ADDRESS, which is used for any further communication. If the server detects some duplicates, it sends a `DYNAMICADDRESS_TYPE_REDO` message to the nodes that selected this particular address. These nodes must randomly select a new address and re-submit it to the server in a `DYNAMICADDRESS_TYPE_VERIFY` message.

Step 4 (multiple servers): In an optional last step, all surrounding servers initiate their distributed detection algorithm. If duplicates are discovered, a `DYNAMICADDRESS_TYPE_REDO2` message is sent to all the nodes with the corresponding address.

If messages get lost, for example the HELLO message or the REDO message, the corresponding node will not be able to assign an address in the current round. It must wait for a subsequent round in which it can participate again at the allocation algorithm.

4. Application Scenario

In order to validate the algorithm, we implemented the dynamic address allocation scheme in the context of a project that focuses on dynamic reconfiguration and reprogramming of sensor nodes using mobile robot systems [23]. Due to the steadily increasing heterogeneity and dynamics in terms of hardware and software configurations in sensor networks, software management is becoming one of

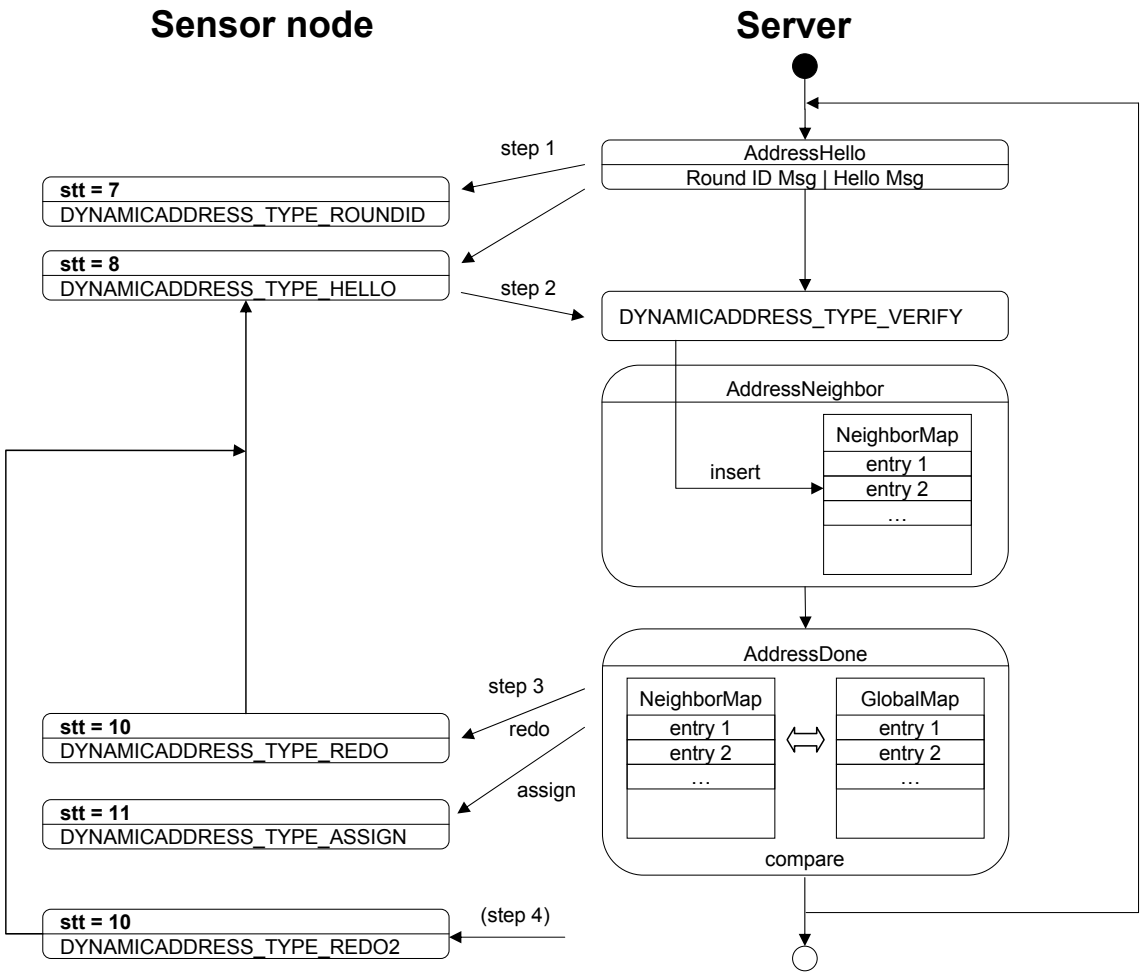

Figure 5. Execution scheme of the dynamic address allocation

the most prominent challenges in this domain. We developed a profile-based software management scheme [19] that consists of a dynamic profile-matching algorithm to identify current SW/HW configurations, an on-demand code generation module, and mechanisms for dynamic network-centric reprogramming of sensor nodes. A mobile robot system is employed for decision processes and to store the source code repository. Our proposed address allocation scheme is used to prevent global pre-configuration of all network nodes.

Figure 6 shows the principal concept of reconfiguration [9]:

a. Depending on the goal, the robot drives to the position in the sensor network where reconfiguration is necessary (we do not assume a particular navigation scheme, various mobility models can be applied).

b. The robot collects information about the environment, builds the context, and explores its neighborhood. In this step, additional actions can be initiated such as preparing the sensor calibration or starting an algorithm for dynamic addressing.

c. All sensor nodes respond to the exploration message by sending their current profiles (HW/SW descriptions).

d. The robot uses the information gathered in steps b) and c) for profile matching and to assign the roles of the sensor notes (depending on the current goal). Finally, it creates the new binaries of the sensor notes.

e. The robot reprograms selected sensor notes over the air.

In our lab, we use the Robertino robot platform developed at the Fraunhofer Institute AIS running embed-

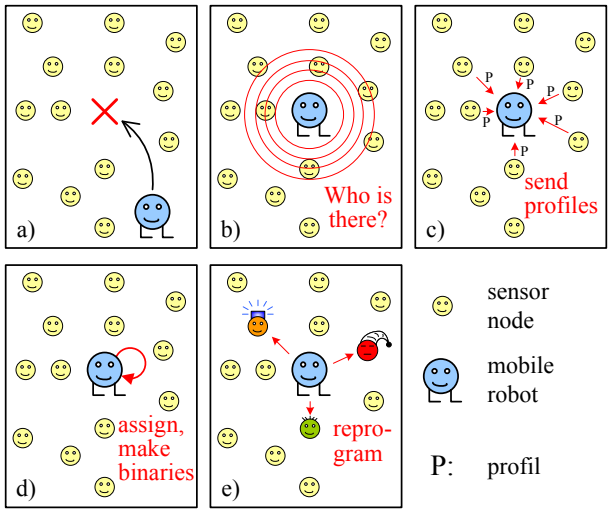

Figure 6. Application scenario for reconfiguration [9]

ded Linux and Mica2 sensor motes developed at UCB running TinyOS. For direct robot-sensor communication, we installed a single Mica2 mote on the robot.

We successfully used the dynamic address allocation scheme for management issues in WSN as described in this paper in order to provide a communication infrastructure for communication between the robot system and surrounding sensors. The implementation allows to choose 16bit addresses as used in the algorithm description.

In our experiments, we never observed the special case of address duplicates. In order to test this case, we manually configured specific nodes with a fixed address. Using this kind of experimentation, we verified the correct behavior of the round based duplicate address detection.

5. Conclusion

In conclusion, it can be said that we developed an address assignment algorithm that works in a localized environment. Therefore, the overhead due to management of topology and uniqueness of the addresses becomes very low. Additionally, the method profits from the single-hop communication that is usually more reliable compared to a multi-hop approach. Basically, we selected particular solutions from PDAD and DHCP to create an efficient and robust dynamic address allocation scheme for management and control in wireless sensor networks.

Compared to PDAD, our solution is more reliable, has less overhead, and is independent of the employed routing algorithm. Similarly to DHCP, the algorithm is server-centric, i.e. the allocation is initiated by a particular system. The duplicate address detection is performed by the same system. Therefore, this verification is very simple. The communication overhead increases linearly with the number of nodes. New nodes do not influence previous allocations. This is also true for waking up nodes performing busy-sleep cycles. They are processed individually in a new sub round.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A Survey on Sensor Networks. *IEEE Communications Magazine*, 40(8):102–116, August 2002.

[2] A. Baggio. Wireless sensor networks in precision agriculture. In *ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005)*, Stockholm, Sweden, June 2005.

[3] C. L. Barrett, S. J. Eidenbenz, and L. Kroc. Parametric Probabilistic Sensor Network Routing. In *International Conference on Mobile Computing and Networking*, San Diego, CA, USA, 2003.

[4] A. Boukerche and S. Nikoletseas. Protocols for Data Propagation in Wireless Sensor Networks: A Survey. In M. Guizani, editor, *Wireless Communications Systems and Networks*. Kluwer Academic Publishers, Date 2004.

[5] C.-Y. Chong and S. P. Kumar. Sensor Networks: Evolution, Opportunities, and Challenges. *Proceedings of the IEEE*, 91(8):1247–1256, August 2003.

[6] F. Dressler. Self-Organization in Ad Hoc Networks: Overview and Classification. Technical Report 02/06, University of Erlangen, Dept. of Computer Science 7, March 2006.

[7] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, March 1997.

[8] R. Droms. Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6. RFC 3736, April 2004.

[9] G. Fuchs, S. Truchat, and F. Dressler. Distributed Software Management in Sensor Networks using Profiling Techniques. In *1st IEEE/ACM International Conference on Communication System Software and Middleware (IEEE COMSWARE 2006): 1st International Workshop on Software for Sensor Networks (SensorWare 2006)*, New Dehli, India, January 2006.

[10] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-Based Ad Hoc Routing. In *IEEE INFOCOM 2002*, pages 1707–1716, June 2002.

[11] X. Hong, K. Xu, and M. Gerla. Scalable Routing Protocols for Mobile Ad Hoc Networks. *IEEE Network*, 16:11–21, July/August 2002.

[12] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen. Scalable Routing Strategies for Ad Hoc Wireless Networks. *IEEE Journal on Selected Areas in Communications: Special Issue on Ad-Hoc Networks*, 17(8):1369–1379, August 1999.

[13] D. Kempe, J. Kleinberg, and A. Demers. Spatial Gossip and Resource Location Protocols. *Journal of the ACM (JACM)*, 51(6):943–967, November 2001.

[14] T. J. Kwon and M. Gerla. Efficient Flooding with Passive Clustering (PC) in Ad Hoc Networks. *ACM SIGCOMM Computer Communication Review*, 2002.

[15] H. Liu, P. Wan, X. Jia, X. Liu, and F. Yao. Efficient Flooding Scheme Based on 1-hop Information in Mobile Ad Hoc Networks. In *25th IEEE Conference on Computer Communications (IEEE INFOCOM 2006)*, Barcelona, Spain, April 2006.

[16] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless Sensor Networks for Habitat Monitoring. In *First ACM Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, USA, September 2002.

[17] S. Pleisch, M. Balakrishnan, K. Birman, and R. van Renesse. MISTRAL: Efficient Flooding in Mobile Adhoc Networks. In *Seventh ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc 2006)*, Florence, Italy, May 2006.

[18] Y. Sun and E. M. Belding-Royer. A study of dynamic addressing techniques in mobile ad hoc networks. *Wireless Communications and Mobile Computing*, 4(3):315–329, April 2004.

[19] S. Truchat, G. Fuchs, S. Meyer, and F. Dressler. An adaptive model for reconfigurable autonomous services using profiling. *International Journal of Pervasive Computing and Communications (JPCC): Special Issue on Pervasive Management*, 2006.

[20] K. Weniger. Passive Duplicate Address Detection in Mobile Ad hoc Networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, USA, March 2004.

[21] K. Weniger. PACMAN: Passive Autoconfiguration for Mobile Ad hoc Networks. *IEEE Journal on Selected Areas in Communications (JSAC)*, March 2005.

[22] K. Weniger and M. Zitterbart. Address Autoconfiguration in Mobile Ad Hoc Networks: Current Approaches and Future Directions. *IEEE Network Magazine: Special issue on 'Ad hoc networking: data communications & topology control'*, July 2004.

[23] Z. Yao, Z. Lu, H. Marquardt, G. Fuchs, S. Truchat, and F. Dressler. On-demand Software Management in Sensor Networks using Profiling Techniques. In *ACM Second International Workshop on Multi-hop Ad Hoc Networks: from theory to reality 2006 (ACM REALMAN 2006), Demo Session*, pages 113–115, Florence, Italy, May 2006.

[24] Y. Yu, R. Govindan, and D. Estrin. Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks. Technical Report UCLA/CSD-TR-01-0023, UCLA Computer Science Department Technical Report, 2001.