Supporting stateless address Auto-configuration in IP-based Wireless Sensor Networks

Shafique Ahmad Chaudhry, Cormac J. Sreenan
{s.chaudhry,cjs}@cs.ucc.ie
Mobile and Internet Systems Laboratory
Department of Computer Science,
University College Cork, Ireland

Abstract

IP-based Wireless Sensor Networks (IP-WSN) are envisioned to play an important role in the realization of pervasive environments. Comprised of hundreds or even thousands of nodes, these networks may never undergo pre-deployment address configuration. The stateless autoconfiguration capability is therefore essential to the spontaneity of such pervasive networks. Recently, several address autoconfiguration schemes have been proposed with their applications in sensor networks as well as in mobile ad-hoc networks. However, we argue that there is no singular scheme that meets the requirements of IP-WSN environments. In this paper, we discuss the requirements for stateless auto-configuration in IP-WSN and present a hierarchical duplicate address detection mechanism for IP-WSNs. In our solution we propose Duplication Address Detection (DAD) Resolver Proxy Agents (DRPA) to act as cache service for DAD resolver to localize the DAD procedure, in order to reduce the DAD overhead traffic. We also propose algorithms to make sure that the node executing DAD procedure is always connected to the nearest DRPA. Analytical results conclude that our architecture and algorithms help reducing the traffic overhead for DAD procedure, decrease the DAD time, and save nodes' energy considerably in IP-WSNs.

Key Words: Duplication address detection, DAD resolver proxy, Neighbor assisted DRPA discovery.

1. Introduction

Wireless sensor networks (WSNs) are on-the-fly networks formed by tiny sensor nodes with highly constrained computation, bandwidth and energy resources. WSNs are envisioned to be connected with each other and with other wired networks in order to maximize the utilization of information and other resources which are mainly associated with IP networks [1]. This integration will help realizing ubiquity by allowing users to access the services across sensor networks and IP networks. The existence of IP-based sensor networks (IP-WSN) brings in generous convenience for the users but implicates underlying requirements and technical challenges for the research community. Address auto-configuration, a widely accepted paradigm is one of the key issues.

Address auto-configuration is essentially required in IP-WSNs because manual configuration is not always a viable option for WSNs, which are considered to have a large number of nodes. Address auto-configuration could be achieved using a stateful or a stateless approach.

Stateful schemes use a cautious approach that pre-allocates a pool of addresses and nodes that join the network must select one of the available addresses. Such an approach falls short of practical utility in situations where the network size can be as large as thousands of nodes. For example DHCP (Dynamic Host Configuration Protocol) [2] which is widely used in IP (Internet Protocol) networks could be considered as a solution. Although the sink may be able to act as a DHCP server, sensor nodes which are not neighbor nodes to the sink have difficulty accessing the DHCP server in a sensor network that uses short range multi hop communication. This is because DHCP requires guaranteed access to the DHCP server from any node by one-hop; i.e., long-range direct communication.

The stateless approach has been envisaged to handle this limitation of the stateful approach. In stateless approaches, the node who wishes to join the network generates an IP-address for itself. However, the addresses must be unique. The addresses must also be properly routable. Examples of such approach are SAA (Stateless Address Auto-configuration) [3] developed for fixed networks and [4], [5] developed for mobile ad hoc networks operated without infrastructural support. A node executes duplication address detection (DAD) procedure to verify that a tentative link-local address is not already in use by another node on the link. After verification, the link-local address can be used permanently by the node and used to form an IP address.

The DAD procedure generally involves broadcast in order to check the uniqueness of the address with the network. These architectures are generally broadcast-based, therefore, exhibit some implicit disadvantages: a) broad-cast systems scale poorly with increased network size, and b) it can use substantial network bandwidth. Furthermore, the address validation time is also non-uniform due to the need for synergistic response from all the network nodes involved in the process. Maintaining a central server as Duplication Address Detection (DAD) resolver inherits the limitations of stateful approach.

978-0-7695-3596-8/09 $25.00 © 2009 IEEE
DOI 10.1109/ITNG.2009.214

We assert that both approaches have shortcomings to be used for IP-WSNs because the address auto-configuration architecture for IP-WSN needs to be scalable and flexible which can handle large number of nodes, but at the same time, this architecture must support localized DAD procedure in order to increase network capacity.

In this paper, we focus on addressing the abovementioned problems by considering the fact that both stateful and stateless schemes have to coexist in order to meet the conflicting design considerations. We present a proxy-based hybrid scheme that combines both the schemes for address auto-configuration in IP-WSNs. Stateless approach is used to make sure that node can generate its address by itself and then can validate it from the closest duplication address resolver proxy. The use of proxies relaxes the notion of having dedicated DAD resolvers in the network, yet localizes the DAD process.

The rest of the paper is organized as follows. Section 2 discusses about the ongoing work in address auto-configuration for ad hoc and sensor networks. In section 3, we present our scheme for address auto-configuration and DAD resolution with a detailed description. In section 4, performance of our scheme is numerically analyzed supported by intuitive logic and discussion. Finally, we present our conclusions and suggest future work.

2. Related Work

Address auto-configuration and duplication address detection have been vastly studied for traditional networks as well as mobile ad-hoc network, however, little has been done for WSNs especially IP-WSNs.

One may argue that that automatic address assignment can be easily achieved by using unique hardware IDs like IEEE MAC address. Although such a hardware address could be unique it cannot guarantee the uniqueness of address [6]. The main reason is that hardware manufacturers may use an unregistered MAC address. In addition failures in the manufacturing process can lead to duplicate MAC address generation. Furthermore, most hardware allows user to change the MAC address to arbitrary values.

Gunes and Reibel in Zeroconf [7] proposed an auto-configuration protocol that utilizes a centralized allocation table. The address agent (AA) which is elected maintains the allocation table which contains previously assigned IPv6 addresses, consequent MAC addresses, and lifetimes etc. AA periodically floods the allocation table to validate nodes in the network. This approach has an overhead due to the periodic flooding in the network which is not feasible for battery and bandwidth constrained IP-WSNs.

Prophet addressing scheme [5] utilizes a stateful function *f(n)* to generate a series of random numbers. The first node let suppose A in the network sets its IP address and choose a random state value as the seed of *f(n)* to compute a sequence of addresses locally for the network. Additional nodes can obtain IP address from A, as well as the state value as the seed for their function *f(n).* Same process continues as the new nodes join the network. The function *f(n)* calculated in such way that the likelihood of the address duplication should be minimal, but still there is an ambiguity of duplication. Although it can detect the duplicate addresses between all the nodes in the network, in order to run DAD procedure, all the network nodes have to transmit and receive a control message periodically. Such continuous communication wastes the sensor nodes' energy.

In Weak DAD [8] a unique node key is incorporated in the routing control packet and in the routing table entries. Two nodes choose same address can be identified by their representative unique keys. It is integrated with the routing protocols and can detect duplicate address detection continuously by checking the routing protocol packets. There is an additional overhead due to enclosure of unique key in the routing protocol control packet

SAA [3] uses a distributed approach and does not rely on central server. The DAD procedure of this approach, however, works only between the neighbor nodes where the link is connected. Therefore it cannot guarantee to detect duplication address detection in the sensor networks.

Motegi et al. [9] have proposed a hierarchical approach for event-driven sensor networks. Though the approach is energy efficient its scope is only for event-driven networks and does not cover the issues associated with IP-WSN.

From the above discussion we assert that it is difficult to use the existing address auto-configuration methods in the IP-WSN. Therefore we propose a hybrid scheme for auto-address configuration in IP-WSN.

3. Proxy-Based Approach

Before outlining our approach, we would like to describe the characteristics we assume in the IP-WSNs:

1) The sensor nodes are randomly switched on, which also means that there is always a possibility of a new node joining the network.
2) Both the radio vicinity and the size of service area is known.
3) The communication links are statistically symmetric.

In our approach, each node self-assigns an IP address, i.e., stateless approach as a local process and then validates this address. A central DAD Resolver (DR), hosted by a suitable node, e.g. the gateway node, could be used to verify the uniqueness of address for every sensor node in the network. As IP-WSNs can have a large number of nodes, placement of a single DR is not a good strategy because: a) the single DR would suffer from a high traffic load from the nodes executing DAD procedure b) the average path length (in hops) between the node running DAD and DR shall be high, and c) the DR advertisement overhead shall be high. The packet delivery ratio and delivery time increases significantly, as the path length increases in terms of hops. Considering all the above mentioned factors, we introduce DAD Resolver Proxy Agents (DRPA) to be deployed in IP-WSN. The DRPA acts as a DAD Resolver cache, within certain proximity, on behalf of the DR.

A DRPA differs from DR in the following ways; a) A DR maintains the address directory for the whole IP-WSN. Whereas a DRPA contains just the address directory cache for the proximity it is deployed in. b) A DR is generally an independent entity which can exist with or without peer DRs, whereas a DRPA is just a proxy, which is dependent on an already existent DR.

Each DRPA cooperates with peer DRPAs in order to help resolving the address conflicts with next closest proximity. No such cooperation algorithm exists in traditional DAD resolver functionality.

The DRPAs reduces the request traffic to the DR as well as reduces the DAD resolving delay for the nodes. The incorporation of proxy-agents softens the requirement of having a dedicated DRs yet reduces the service discovery overhead. We exploit the fact that IP-WSN nodes are inexpensive by deploying multiple DRPAs, each responsible for a certain area, within an IP-WSN. For example if a city is considered as a large IP-WSN network, a block can be considered as single proximity, which could be managed by placing a DRPA in it. Each DRPA is responsible for maintaining Local Cache (LC), for the addresses within its local proximity, and External Network Cache (ENC) which represents the addresses with the neighbor IP-WSNs.

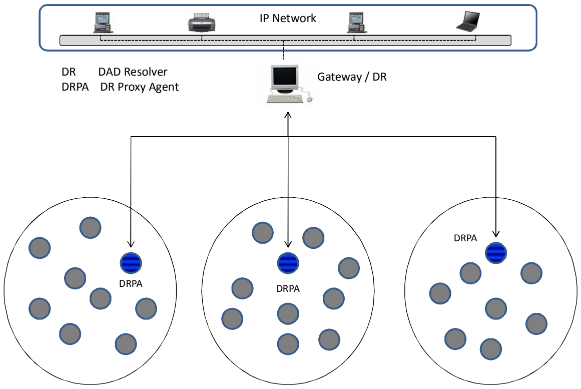

Figure 1. DRPA based architecture

As shown in figure 1, a huge IP-WSN network has been divided into three proximities by deploying one DRPA in certain area. These DRPA are connected to the external IPv6 network through a DR deployed on the gateway. The scope of each DRPA is limited to certain proximity and DAD resolving service is offered by that DRPA in a specified region. It means that DRPA needs to maintain a comparatively small cache. The smaller is the cache size, faster is the conflict search which reduces the address conflict resolution time. All the nodes within the specific proximity register their addresses with the DRPA. The DRPA periodically broadcasts, within its reason, an advertisement message to notify its existence. As the DRPA are deployed at specific locations, the service information maintained by them is also proximity-based.

These DRPAs could be arranged in a hierarchical way i.e. they can communicate with their peer DRPAs as well as with the central DR which might be the part of external IP-based network. This communication helps DRPA maintaining an updated and consistent ENC. DRPAs share their caching information with each other periodically. This sharing of information allows knowing about the addresses registered with the neighbor DRPAs. This periodic sharing of information reduces flooding which, otherwise, will be required to find duplicate addresses from the neighboring proximities.

The basic idea for the node who wishes to run DAD procedure is to connect with the closest DRPA from the user. We support that feature we propose neighbor assisted DRPA discovery protocol to make sure that the new node connects with the closest DRPA.

3.1 Neighbor assisted DRPA discovery protocol

Whenever a node wants to join an IP-WSN, it first tries to discover an existing IP-WSN. IEEE 802.15.4 specifies active and passive scanning procedures for this discovery operation. By following either one of the scanning procedures, the new device determines whether there is a Low Power Wireless Personal Area Network (LoWPAN) in its personal operating space. Once a LoWPAN is found, next step is to assign itself an address and check the validity with the closest DRPA using neighbor assisted DRPA discovery protocol. To implement our protocol, we have defined DRPA Discovery Request (DDREQ) and DRPA Discovery Reply (DDREP) messages. The messages are used as follows:

- *DDREQ*: This message is used to ask the neighbors about their respective DRPAs. Initiated as a one hop-broadcast by the node that needs to find the closest DRPA.
- *DDREP*: This is the reply message in response to a DDREQ. It contains the address of the DRPA as well as distance to the DRPA in terms of hop count.

When a nodes needs to send a DAD request (DREQ) to DRPA, it checks with its single hop neighbors, by broadcasting DDREQ in one hop, the closest DRPA in terms of hop count. The neighbors reply with DDREP that contains the address of closest DRPA and distance to it in hop count. The nearest DRPA, in terms of hop count, is considered as the closest DRPA. In case no neighbor sends a positive DDREP, it means that no neighbor is aware of the DRPA in which case the DDREP can be broadcast in two hops. Another alternative is to re-broadcast the DDREQ after waiting a certain time.

Neighbor assisted DRPA discovery algorithm helps in handling mobility of the new nodes as well. This protocol makes sure that the node stays connected with the existing DRPA as long as it is the nearest one, even when the node is moving. Fig. 2 illustrates the protocol.

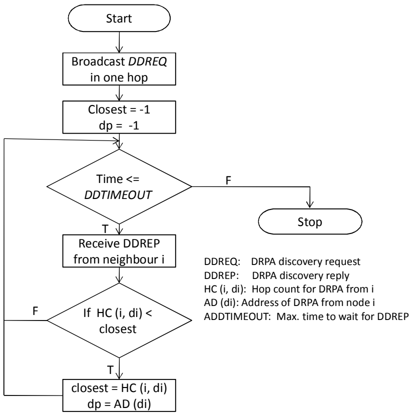
Start
Broadcast DDREQ in one hop
Closest = -1
dp = -1
Time <= DDTIMEOUT
F
Stop
T
Receive DDREP from neighbour i
DDREQ: DRPA discovery request
DDREP: DRPA discovery reply
HC (i, di): Hop count for DRPA from i
AD (di): Address of DRPA from node i
ADDTIMEOUT: Max. time to wait for DDREP
F
If HC (i, di) < closest
T
closest = HC (i, di)
dp = AD (di)

Figure 2. Neighbor assisted DRPA discovery

3.2 Cache Cooperation

As a DRPA is responsible to manage the directory services in a small region, the LC may not be able to entertain all the DAD requests received by the DRPA. The ENC is used to cache the service information of the neighbor networks, i.e., the neighbor IP-WSNs and IP networks. The ENC can be managed through DRPA cooperation, i.e., service information exchange between the DRPAs.

The mechanism for DRPA cooperation is very important factor which affects the DAD resolution time and DAD success ratio. The cooperation between DRPAs can be implemented using an on-demand model or a proactive model.

In the on-demand model the address directory information is shared with a neighbor only when it requests for it. In this case the ENC is filled in increments. This approach generates less traffic overhead but increases average DAD response time because a large number of DAD requests could be forwarded to the neighbor DRPAs. Moreover, ENC is not refreshed periodically, generating a high probability that the address directory information is stale.

In the proactive model the DRPAs periodically exchange the directory information with their neighbors. The DAD success ratio depends heavily on cache size, ENC hit ratio, and information exchange interval. But this periodic information exchange means more overhead traffic.

3.3 Placing DPAs Optimally

DRPAs can be placed in the network using any criteria. The easiest way probably would be to deploy them uniformly within the network area. However, the DRPAs deployment can be made optimal by defining an objective function and the system constraints and finding a solution using a linear programming technique.

In this section we present a placement heuristic to determine the minimal number of DRPAs in an IP-WSN in order to reduce communication overhead by allowing sensor nodes to register to the nearest DRPA with a maximum of 3 hops away. Therefore, the allocation of DRPA to client nodes is a distance-limited problem in the simplest form. Other important considerations include:

C1. Each node must be associated with a DRPA considering that it shall send a DAD request.
C2. Each node must be allocated one and only one DRPA. Such a strict association is only forwards not backwards.
C3. Each DRPA may serve an arbitrary number of nodes. This is the restatement of the later part of C2.

It may be noted that C1 transforms the placement (locating) problem of the DPAs within the sensor network from maximum coverage problem to mere set covering problem [10]. Furthermore, C3 allows us to look for a limited number of DRPAs that can be allocated to all the nodes (that shall need to execute DAD) in IP-WSNs. Using the preceding aspects we can develop a DRPA placement heuristic to determine such an optimal number of DRPAs. The inputs for the model can be expressed as:

Inputs

$$a_{ij} = \left\{ \begin{array}{ll} 1 & \text{if DRPA at sensor node } j \text{ is} \\ & \text{at most 3 hops from node } i \\ 0 & \text{if not} \end{array} \right\}$$

Decision Variables

$$X_j = \left\{ \begin{array}{ll} 1 & \text{if we locate the DRPA at} \\ & \text{sensor node } j \\ 0 & \text{if not} \end{array} \right\}$$

The objective function can be given as:

$$\textit{Minimize} \sum_j X_j \qquad \text{(a)}$$

Subject to

$$\sum_j a_{ij} X_{ij} \geq 1 \qquad \forall i \qquad \text{(b)}$$

$$X_j = 0,1 \qquad \forall j \qquad \text{(c)}$$

4. Analytical Evaluation

To discuss and analyze the performance of DRPA based architecture, we choose DAD procedure overhead, DAD procedure delay, and bandwidth usage.

4.1. DAD Procedure Completion Time

We define the DAD procedure completion time as time from generating an address till its uniqueness is validated. We consider the delay incurred in single-DR, without DR, and DRPA cases.

In a central DAD-resolver-based solution, the new node that wishes to join the network can find the DR, using active or passive discovery. If the new node does not have a route to DR, then it must find one to the DR. Once the path to DR is known, the latency to complete a DAD procedure is dependent on the delay to reach the DREQ from the node to DR, the processing time at DR, and the time taken by DREP from DR to the node. This delay mainly consists of network delay, i.e., routing and forwarding delays, plus the request processing delay at the DPA.

If h is the average path length between the node and DR, and D_p is the time to process a request at DR the average time to get a service reply from a DR is given by:

$$T = 2\left[\frac{1}{R/L-\lambda}\times(h-1)+D_t\times(h-1)\right]$$

Where L is the average message length,   is the packet generation rate (messages per second) and R represents the data rate. The symbol D_t represents the average transmission delay per node.

In case no central DR exists, the node broadcasts its generated address in the whole network to validate its uniqueness. The delay depends on the network size, the average path length and average number of neighbors.

For DRPA based DAD resolution the neighbor assisted DR discovery considerably reduces the connection time to DR. In the same way as there are multiple DRPAs available in the network, average path length is reduced which also decreases the DAD completion time considerably. Figure 3 shows MATLAB graphs for the comparison between all these approaches.

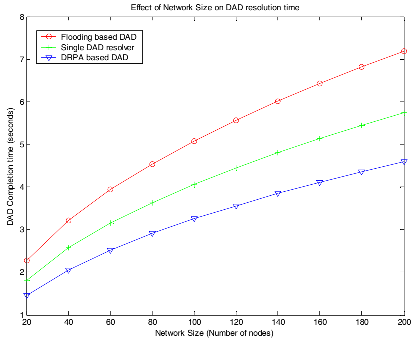

Figure 3. DAD Procedure completion time

As we can see that DRPA-based scheme performs much better. The main reason is that in DRPA based scheme the one-hop neighbors provide the DR information, after which only unicast communication is needed between DR and new node to check the address duplication.

4.2. DAD Completion Overhead

We define DAD completion overhead as the total number of packet generated with the network for single DAD procedure.

At first, we just consider a single DR case. The DAD completion overhead is the sum of traffic generated to find the path to the DR, plus ADREQ and ADREP packets generated within the network. The number of packets generated for finding the path to the DR depends on the DR discovery and the routing scheme being used.

In broadcast based approach the overhead depends on the average number of hops per path and average number of neighbors per node.

In DRPA based scheme the overhead to discover the DRPA is reduced significantly through neighbor assisted DRPA discovery. In case there are *x* DRPAs in the whole network the total DAD procedure overhead in term of packets is the sum of DAD procedure overhead for all the DRPAs, plus the information exchange between the DPAs, plus the communication cost between DRPAs and DR. If O_{DXA} and O_{PXA} represent the average number of packets sent by each DRPA to its DR and peers respectively, then the total packet overhead for the system can be calculated as:

$$O_{ToTAL} = \sum_{i=1}^{x} O_{DRPA} + O_{DXA} + O_{PXA}$$

The total bandwidth utilization for the whole system could be calculated by placing the average sizes for the messages used in the process in respective equations. Figure 4 shows the significant results for using DRPA approach. The main reason is that DRPA mitigates the broadcast storms which are resultant of a DAD procedure in a pure distribute (broadcast-based) environment.

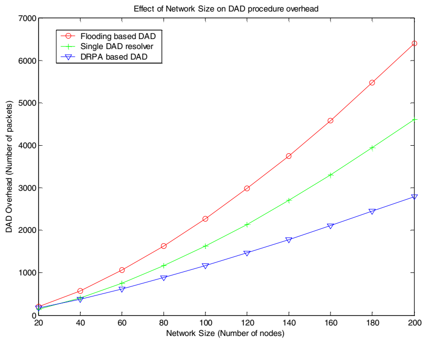

Figure 4. DAD overhead comparison

4.3. Energy Consumption

The physical energy model can be divided into two parts, i.e., reception and transmission. The fact that more traffic is generated over the network means that nodes shall consume more energy.

In case of DR based scheme, the energy consumption of nodes depends hugely on the DR advertisement frequency and DAD execution frequency in the network. For the broad-cast DAD procedure where no central server is available, the energy consumption depends on the frequency of DAD requests. The reception energy can be calculated as $Er = V * Ir * t$ where V, Ir, represent radio chip supply voltage, reception current as given in [11]. The time t can be calculated by multiplying the packet length and the inverse of the radio chip bit rate. The transmission energy can be calculated in the similar fashion except different levels of transmission consume differently.

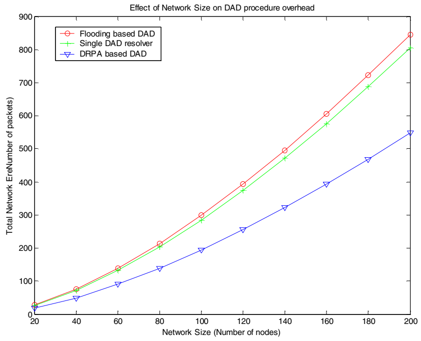

Figure 5. Energy consumption

Figure 5 shows the average energy usage comparison using all the three approaches. It is obvious that the energy uses in DRPA is less than both the other approaches because of the less traffic generated for establishing a path to DRPA. It should also be noted that the DAD procedure overhead as well as energy consumption can be higher in DR as compared to distributed approach if the DR advertisement frequency is equal to or higher than DAD request frequency.

5. Conclusion and Future Work

In this paper, we have presented a hierarchical proxy-based approach to support stateless address auto-configuration in IP-WSNs. Not only our scheme relaxes the condition of having dedicated DRs in the network, it also localizes the DAD procedure communication considerably reducing the overhead traffic and DAD delay. In future works, we plan to test the functionality of the proposed solution and test it in real environment.

Acknowledgement

This work was performed as part of Embedded Networked Sensing (EmNetS) project funded by Enterprise Ireland's industry lead Wisen Program.

References

[1] Transmission of IPv6 Packets over IEEE 802.15.4 Networks, IETF, RFC 4944, 2007.

[2] R. Droms, "Dynamic host configuration protocol," IETF RFC 2131, 1997.

[3] T. Narten, "Neighbor discovery and stateless autoconfiguration in IPv6," IEEE Internet ComputingMag., vol.3, no.4, pp.54–62, 1999.

[4] C.E. Perkins, J.T. Malinen, R. Wakikawa, E.M. Belding-Royer, and Y. Sun, "IP address autoconfiguration for ad hoc networks," IETF Internet Draft, 2001.

[5] H. Zhou, L.M. Ni, and M.W. Mutka, "Prophet address allocation for large scale MANETs," Proc. 22nd Annual Joint Conf. of IEEE Computer and Communications Societies (INFOCOM), 2003.

[6] K. Weniger, M. Zitterbart, and U. Karlsruhe, "Address autoconfiguration in mobile ad hoc networks: Current approaches and future directions," IEEE Netw. Mag., vol.18, no.4, pp.6–11, 2004.

[7] M. Gunes, J. Reibel, "An IP Address Configuration Algorithm for Zeroconf Mobile Multihop Ad Hoc Networks," Proc. Int'l. Wksp. Broadband Wireless Ad Hoc Networks and Services, Sophia Antipolis, France, Sep. 2002.

[8] N. Vaidya, "Weak Duplicate Address Detection in Mobile Ad Hoc Networks," Proc. ACM MobiHoc 2002, Lausanne, Switzerland, pp. 206–16, June 2002.

[9] S. Motegi, K. Yoshihara, H. Horiuchi, "Address auto-configuration for Event-Driven Sensor Network," IEICE Transactions on Comm., Vol.E-88B, No. 3, pp 950-957, 2005.

[10] Mark S. Daskin, "Network and Discrete Location; Models, Algorithms and Applications," John Wiley and Sons, Inc. 1995.

[11] Chipcon AS, CC2420 Datasheet