Infrastructure Optimization of Flight-Formation Inspired Self-organization for Address Configuration in Sensor Networks

Rui Teng, Bing Zhang, and Jian-Qin Liu

National Institute of Information and Communications Technology, Japan
`{teng,zhang,liu}@nict.go.jp`

Abstract. Self-organized sensor networks are expected to automatically configure sensor nodes into networks. Node addresses are essential for network communications, and the address size has a substantial impact on the amount of energy consumption during the long-term report of small sensing data. This paper introduces a Scalable and Dynamic Infrastructure based Configuration scheme (SDIC), which addresses the problem of how to organize a large number of sensor nodes to configure small addresses with scalability and energy efficiency. The special features that make our approach unique are the exploitation of an optimized autoconfiguration infrastructure that is inspired by the flight formation of migration birds. SDIC enables sequential address assignment in a deterministic manner without address conflicts. SDIC utilizes mechanisms of the optimized server term control to achieve the scalability of infrastructure and configuration operation. The evaluation results of SDIC show that it achieves small-size address with few conflicts and with energy efficiency.

Keywords: Self-organization, bird flight formation, address autoconfiguration, network infrastructure.

1 Introduction

Two features distinguish the sensor networks with the conventional Internet. One is the self-organization of sensor nodes. The other is energy efficient operations in the network. Small sensor nodes are deployed in a region of interest to observe physical phenomena such as temperature, humidity, or solar radiation. Battery-powered sensor nodes are expected to self-organize into networks and provide long-term operations of reporting data and keeping network connectivity [1],[2].

The IP addresses have a high energy cost for sensor nodes. It is possible for a sensor node to adopt a local unique address as its network-layer address, because delivery and process of sensing data in a sensor network are generally carried out inside a task area, such as a monitoring region [3]. In other words, each node in a distributed system can have a node address that is unique with respect to the connectivity of the network. A network-wide local unique address has a potentially

J. Suzuki and T. Nakano (Eds.): BIONETICS 2010, LNICST 87, pp. 587–600, 2012.

short address size. For instance, a typical sensor network of a few hundred nodes could use as few as 10 bits if the addresses were locally unique. Although sensor attributes, such as temperature, humidity and location, can be used to partially identify the sensor destinations in an information query, attribute addresses can not distinguish individual sensor node and the sources of sensing data.

To configure addresses for a large number of nodes, it is difficult to manually allocate the addresses to a number of nodes in sensor networks, and a reasonable option is to use autoconfiguration of sensor addresses. The address autoconfiguration is a self-organization procedure. It selects, allocates, and assigns a unique network address to each unconfigured node. However, most of conventional autoconfiguration schemes deal with the IP address configuration, generally within a large address space [4],[5],[6],[7],[8],[10],[12],[13]. Although a few schemes of Non-IP address autoconfiguration have also been adopted to configure addresses, these configuration schemes either assume using a pre-defined large address space for small scale nodes, or configuring the addresses that are not unique in the network [9], [11]. To configure small-size addresses of sensors, it requires that addresses being assigned within a potential smallest address space for a number of sensor nodes. This leads to a high probability of address collisions.

This paper introduces a Scalable and Dynamic Infrastructure based Configuration scheme (SDIC), which self-organizes sensor nodes to efficiently configure addresses in a small address space. SDIC utilizes the model of the birds' flight formation, in which all birds shares the responsibility of being lead bird by rotation to keep the team consistency and have energy saving [15]. Like the role rotation among birds in flight formation, SDIC constructs a controllable network infrastructure by utilizing dynamic roles of address server and address client among sensor nodes. The address servers manage the use of address space in a deterministic manner and assign address in a sequential way. Each address server only maintains small amount of state information (such as the address sequence) during the configuration. SDIC utilizes the optimized control of the server term that manages the role rotation of sensors to avoid overuse of energy at a certain node. As a result, SDIC has a temporary centralized infrastructure, which, however, operates in distributed manner for the long-run.

The rest of the paper is organized as follows: Section 2 presents related work. Section 3 describes the flight formation inspired infrastructure self-organization in SDIC scheme. Section 4 describes the basic SDIC autconfiguration. Section 5 describes the optimized control of address-server's term. Section 5 presents the simulation results of the proposed SDIC protocols. Section 6 concludes.

2 Background and Related Work

Conventional approaches of address autoconfiguration can be categorized as infrastructure-based or infrastructureless-based according to the utilization of predefined network infrastructure. In traditional IP networks, the address configuration is infrastructure-based due to the use of address servers. Dynamic Host Configuration Protocol (DHCP) operates based on centralized server-client architecture, is an effective address assignment approach in the IP Internet [8].

In mobile ad hoc networks, address configurations are infrastructureless: without the dependence of predefined address server. Autoconfiguration of node addresses in ad hoc networks has been extensively studied in recent years. There are mainly two types of address autoconfiguration in ad hoc networks, stateful autoconfiguration and stateless autoconfiguration. Stateful address autoconfiguration adopts address allocation table to maintain uniqueness of addresses, such as MANETconf [4], prophet [13]. Stateless autoconfiguration utilizes the on-demand duplicate address detection, such as WDAD, IETF zero-configuration [5],[14]. Many of them are based on Duplicated Address Detection (DAD) approach, which operates based on full distributed self-organization architecture [6]. The existing address autoconfiguration approaches are generally utilized for IP address configuration (for the ease of installation) in a relative small network. These approaches have a general target of dealing with the invalid address configuration caused by the network merge and partitions [4],[5],[7]. But such a target is not the ultimate goal in sensor networks, the topologies of which are generally static and energy efficiency is of more importance than the mobility issue.

A few approaches of address autoconfiguration have been proposed in sensor networks. Random addressing lets each sensor node select an address randomly in a defined address space. Random addressing is a simple way for each node to get an address. It is an infrastructure-less approach and operates in a full distributed organized manner. But the uniqueness of the address can not be guaranteed, and the incidence of address collisions will be high when the address size is short [9]. Treecast addressing has also discussed node address issues in sensor networks [10]. It is an infrastructure-based approach, however, it utilizes a static centric organized architecture assuming there are special sink servers already deployed in sensor networks. Furthermore, Treecast addressing does not consider the energy efficiency aspect for node addressing in sensor network, such as configuring small-size addresses. A MAC addressing scheme has been proposed in [11], which achieves small MAC address by adopting address reuse in the networks. However, the proposed approach cannot be utilized for configuring unique network-layer addresses. [16] has proposed energy-efficient MAC and network-layer addressing schemes in sensor networks. However, it have not addressed and analyzed the efficient organization of sensor nodes and the infrastructure optimization for address configuration. Spatial-time based address configuration was proposed in [17], in which the location of nodes and time of configuration are used to represent the node address. Due to the large configuration space of location and time, there are few address conflicts. However, the spatial-time configuration has not considered the problem of large address size caused by the using location and time in the address.

3 Flight-Formation Inspired Network Infrastructure Self-organization

3.1 Network Model

A sensor node consists of sensing, computing, and communication capabilities. The node address of a sensor is configurable in that the address and its size can be set up in an on-demand manner rather than being predefined before the node joins the network. Unlike the conventional Internet nodes, the sensor node generally has a power constraint.

A self-organized sensor network does not depend on the predefined network infrastructure such as network servers. In contrast, sensor nodes cooperatively perform the role of network infrastructure, automatically configuring sensor nodes into a network. A typical self-organized sensor network operates as follows. Sensor nodes are firstly deployed in a field with a certain density that enables nodes to be connected, and sufficiently cover the sensing area. Then, the nodes, which operate in an unattended mode, self-organize into a network. Node addresses are essential for nodes to cooperatively form the network, and communicate with each other. After the configuration, sensor nodes start to collect and report the information about their surroundings. Each sensor node connects to others by either one hop connection or multi-hop connections.

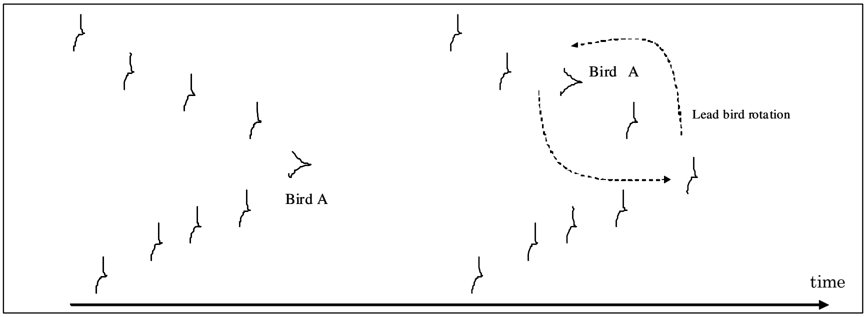
The task of address autoconfiguration is to assign each node a unique address in an automatic manner. To configure small-size unique addresses, it requires configuring addresses within potential smallest address space for a number of sensor nodes. However, it is difficult to utilize the individual operation of node configuration such as distributed random address selection and assignment, which is widely used in conventional autoconfiguration schemes. This is because there will be many collisions to configure a number of addresses in a small address space. The cooperation of self-organized sensor nodes is necessary to efficiently manage the address space.

3.2 Flight Formation Model

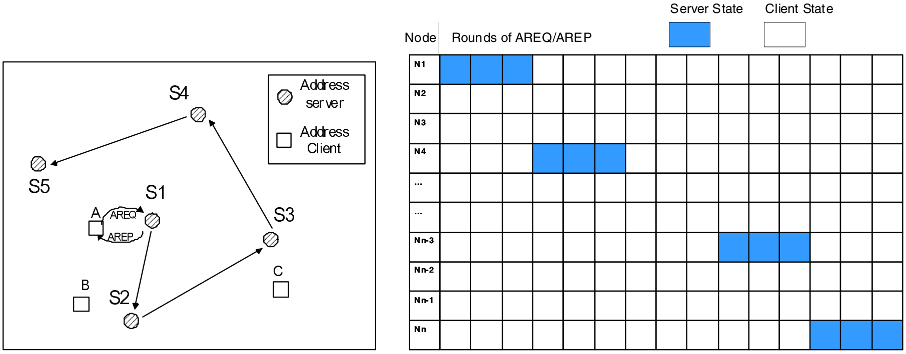
The self-organization for sensor nodes to cooperatively configure addresses is inspired by the formation flight model of birds. Many species of large birds, such as goose, duck, and other migration birds, fly together in formation. A group of such birds will typically fly in a large “V” type [15]. In the “V” type formation, generally there is one bird in the lead position at the vertex of “V”, and others trail behind in two lines, as shown in Fig.1 (a).

Researchers who have studied formation flight believe that birds fly in this way for two reasons. The first reason is that the shape of formation reduces the drag force that each bird experiences compared to it flying alone [18]. The second reason that may explain the flight is that the “V” formation allows the birds to communicate more easily with good visual contract of each other.

The formation flight keeps the consistency of speed and direction of birds as a group. Each bird takes the advantages of the power of the formation. But the lead bird in the vertex of “V” consumes more energy than others. When the lead bird gets tired, it will rotate back in the wing, and another bird takes the lead position, as shown in Fig.1 (a). This results in the dynamic formation of “V” to keep the energy balance among birds [15].



(a) Flight formation of birds



Node															
N1	🟦	🟦	🟦	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
N2	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
N3	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
N4	☐	☐	☐	🟦	🟦	🟦	☐	☐	☐	☐	☐	☐	☐	☐	☐
...	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
...	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Nn-3	☐	☐	☐	☐	☐	☐	☐	☐	☐	🟦	🟦	🟦	☐	☐	☐
Nn-2	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Nn-1	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Nn	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	🟦	🟦	🟦

(b) Address configuration with server rotation

(c) Server term control

Fig. 1. Overview of SDIC address configuration. The bird flight formation model describes that all birds share the responsibility of being lead bird by rotation to keep the team consistency and to have energy balance. This brings insights into the self-organization of sensor networks for address autoconfiguration.

3.3 Insights from Flight Formation to SDIC

The bird flight formation model brings important insights into the self-organization of sensor networks for address autoconfiguration. First, flying birds as a group represent a well established team with shared goal (such as flight migration). With the flight formation, the team operates in a consistent manner with regard to speed and direction. Further, the formation result in the energy saving of birds during the flight. Second, to maintain the flight formation, each bird in the flight has its own roles and responsibilities for the entire group. The lead bird takes most of the responsibility during the flight, but other birds also become leaders when needed.

In SDIC configuration, a mutlihop server-client formation is self-constructed by sensor nodes. The server node manages the use of address space and assigns address to client sensors in a deterministic manner and keeps the consistency of address usage. This formation allows sensor nodes to configure address with little address conflicts, keeps the address assignment within the smallest necessary address space.

The sensors take advantage of the server-client formation with a high successful configuration rate, and save the configuration energy by using easy communication of unicast between server and clients.

In SDIC, like the flight formation, each sensor node has a role of becoming address client or address server. A node in the server state acts as an address server, and then takes the responsibility of address assignment, while a node in the address client state might issues the configuration request or relay packets for other sensors.

Like the lead bird in flight formation, the server node consumes more energy than other nodes. Thus it will rotate back to client after a certain term of server operation. In this paper, the optimized server term is studied by both of analysis and simulation. The server rotation avoids the overuse of energy at one node, balancing the energy consumption of the network. Further, server rotation provides the configuration scalability in the network.

4 Fundamental SDIC Address Autoconfiguration

SDIC configures addresses in a potential smallest address space by assigning address sequentially from low to high without the overuse of address space. The sequential assignment of node addresses requires a deterministic operation rather than an opportunistic operation so as to maintain the consistency of address configuration in self-organized sensor networks.

To realize these functions, SDIC scheme self-organizes sensor nodes into a *dynamic* server-client based network infrastructure. Each address server is automatically elected among sensor nodes, and has a limited server term, after which, another sensor node will be selected as a new server. As shown in Fig.1 (b), the address server performs the address assignment in its term to the address clients, which are nodes that have not had their address configured yet. In the configuration, an address client (e.g. node A) that asks for an address sends an *address-request* (*AREQ*) message to the server (e.g. S1). In response to the *AREQ* message, the address server selects an unused address in the address space and sends an *address-reply* (*AREP*) message, which includes the assigned address, to the address client.

At the beginning of configuration, when a node needs an address to be configured and does not have any information about the address server, the node searches for the address server by broadcasting a server-query message at a random time. If an address server already exists, the server should inform of its presence to the node that is searching for a server. If a searching node receives no reply within a certain period of time, it turns into a server initiation state, assigns itself as an address server, and then announces its status to other nodes by broadcasting a *server-announcement* message. Each sensor node that receives the server-announce message records the server address and also the source route to the address server for future connections. Thus, the route for an address client to deliver an *AREQ* message to an address server is obtained from the source route of a *server-announcement* message. And the route for the server to deliver an *AREP* message to an address client is obtained from the source route of the *AREQ* message from the address client.

SDIC utilizes *the server term control* to obtain the infrastructure scalability. The *server term control* computes and sets an optimized term for each address server. It controls the server rotation (as shown in Fig.1 (c)) among sensors to achieve both deterministic and distributed operation, leading to the server robustness by avoiding the overuse of server energy. A node in a server state perform the server role in a server term that is measured by a certain number of AREQ/AREP rounds in the configuration.

With the central control of address servers, node addresses can be assigned in a sequential manner to sensor nodes. This results in the smallest necessary address space and few address conflicts during the configuration procedure. That is, the address server assigns each address in a sequential increasing order, beginning at 0. A sequence of addresses is maintained at each server, and the address sequence is delivered to a new server from the server that finishes its server term.

5 Optimized Address-Server Term Control

Note SDIC sets each address server a server term. As shown in Fig.2, after an address server (such as Sn in the figure) has assigned addresses n times, it selects a new server from the clients that obtained its address in its server term. In this paper, the last client that is assigned with an address from current address server will be the candidate of new address server (e.g. Sn+1 in the figure). After a new server is selected, the address sequence is delivered from the current server to the new server so as to keep the consistence of address configuration. A simple procedure of server rotation can just includes the new server assignment by delivering the address sequence message to the next server.

There should be a certain condition with regards to the server numbers, by adjusting which the energy consumption at the servers could be optimized. We calculate the server number that results in the minimum energy consumption at a server since the server generally consumes the largest part of energy.

At first we calculate the configuration cost in a centric address configuration operation without server rotation. We define following parameters.

N	The total number of nodes in the network
K	The number of address servers during the configuration
E_S	Energy consumption at an address server
E_{AC}	Energy consumption at an address client
E_{SI}	Energy consumption of address server initiation
E_{SC}	Energy consumption of the address server configuration
E_{SR}	Energy consumption of the address server rotations
E_{TX}	The energy consumption of transmitting a packet
E_{RX}	The energy consumption of receiving a packet

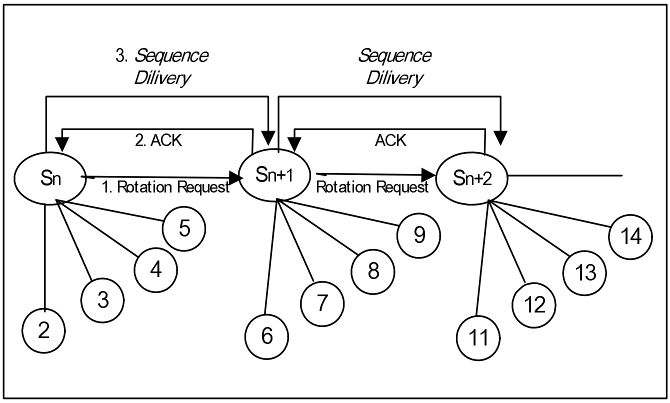


Fig. 2. Server term control. After an address server finishes its term, a new server will be selected to assign addresses to sensors.

The energy consumption at a server, $E_S[i]$, includes three parts: the energy consumption E_{SI} in the server initiation, E_{SC} in the address configuration, and E_{SR} in the server rotation. It can be computed as

$$\begin{aligned} E_S[1] &= E_{SI} + E_{SC} + E_{SR} \\ &= E_{TX} \times 2 + (E_{RX} + E_{TX}) \times [\frac{N-1}{K}] \\ &+ (E_{RX} + E_{TX}) \times (K-1) \end{aligned} \tag{1}$$

$$\begin{aligned} E_S[i] &= E_{SI} + E_{SC} + E_{SR} \\ &= E_{TX} \times 1 + (E_{RX} + E_{TX}) \times [\frac{N-1}{K}] \\ &+ ((E_{RX} + E_{TX}) \times (K-1) + E_{RX}) \end{aligned} \quad i>1; \tag{2}$$

where k is the number of servers during the configuration, each sever assigns (N/K) addresses. In this paper, we adopt the simple procedure of server rotation which just includes the new server assignment by delivering the address sequence message to the next server, and requires the new server to initiate an announcement broadcast. The server announcement lets other node the new server's ID and route so as to avoid much overhead of broadcasting for address clients to find server. The minimum energy consumption at a server can be expressed as follows.

$$\begin{aligned}\min(E_S[i]) &= \min(E_{SI} + E_{SC} + E_{SR}) \\ &= \min(E_{TX} \times 2 + (E_{RX} + E_{TX}) \times [\frac{N-1}{K}] \\ &+ (E_{RX} + E_{TX}) \times (K-1))\end{aligned} \qquad i=1; \quad (3)$$

$$\begin{aligned}\min(E_S[i]) &= \min(E_{SI} + E_{SC} + E_{SR}) \\ &= \min(E_{TX} \times 1 + (E_{RX} + E_{TX}) \times [\frac{N-1}{K}] \\ &+ ((E_{RX} + E_{TX}) \times (K-1)) + E_{RX})\end{aligned} \qquad i>1; \quad (4)$$

To derive the optimal server number K that yields the minimum $E_S[i]$, we calculate the derivative of $E_S[i]$ with respect to K as follows.

$$\begin{aligned}\frac{d}{dk}E_S[i] &= \frac{d}{dk}(E_{SI} + E_{SC} + E_{SR}) \\ &= (E_{RX} + E_{TX}) \times (-\frac{N-1}{K^2}) + (E_{RX} + E_{TX})\end{aligned} \qquad (5)$$

When $K = [\sqrt{N-1}]$, $\frac{d}{dk}E_S[i] = 0$ and the $E_S[i]$ has the smallest value. The corresponding server term is $[\frac{N}{K}]$. Thus, substituting $K = [\sqrt{N-1}]$ to equations (3) and (4) gives the minimum energy consumption at a server.

$$\begin{aligned}&\min(E_S[i]) = E_{TX} \times 2 + (E_{RX} + E_{TX}) \times [\sqrt{N-1}] + \\ &(E_{RX} + E_{TX}) \times ([\sqrt{N-1}] - 1))\end{aligned} \qquad i=1; \quad (6)$$

$$\begin{aligned}&\min(E_S[i]) = E_{TX} \times 1 + (E_{RX} + E_{TX}) \times [\sqrt{N-1}] + \\ &(E_{RX} + E_{TX}) \times ([\sqrt{N-1}] - 1)) + E_{RX}\end{aligned} \qquad i>1; \quad (7)$$

As for the address clients, the corresponding energy cost is as follows.

$$\begin{aligned}&\min(E_{AC}) = 2 \times (E_{RX} + E_{Tx}) + \\ &(E_{RX} + E_{TX}) \times ([\sqrt{N-1}] - 1)\end{aligned} \qquad (8)$$

From the above analysis, we find that there is an optimized server number that gives rise to the lowest energy consumption at servers, leading to both small and balanced energy cost among sensor nodes.

6 Simulation Evaluation

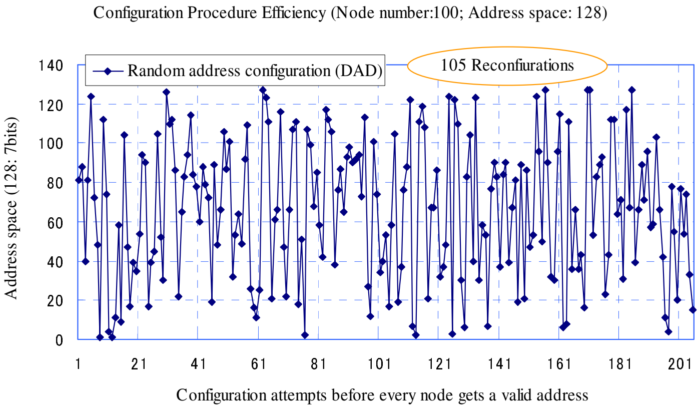
6.1 Simulation Setup

Unless otherwise stated, the simulation is setup as follows. The number of sensor nodes varied from 20 to 256. Each sensor is equipped with a radio module that has a transmission range of 10m. The sensor nodes are distributed over the rectangle fields: field of 100m×20m for 20 nodes, 100m×50m for 50 nodes, 100m×100m for 100 nodes, 100m×150m for 150 nodes, 100m×200m for 200 nodes, and 100m×250m for 250 nodes. Each node boots at a random time within 0-300seconds. Each node has a 32 bits unique temporary address, the difference of temporary address, temporary configured addresses, and configured addresses are identified by the packet type field in the packet header. As for SDIC used in the simulation, each server rotates its role to another node after it has assigned addresses 5 times. That is, a default number of addresses assignment in a round service of one address server is 5. The configuration message is assumed with the same packet size. The proposed SDIC scheme is evaluated and compared with a random address configuration (DAD) approach, because DAD is a widely used approach for address autoconfiguration in self-organized networks[4], [5], [6]. In DAD, each node configures address individually by randomly selecting a candidate address for configuration, and then detects its validity in the network. If the selected address is duplicated with other nodes, the node should configure again by selecting another candidate address.

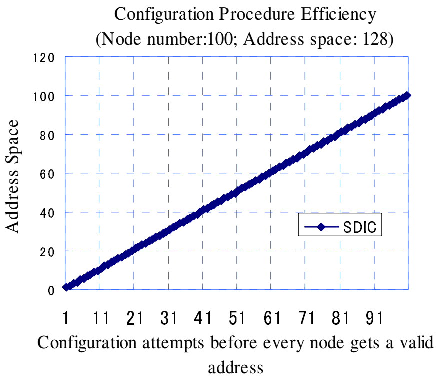
6.2 Numerical Results

A. Configuration Procedure Efficiency

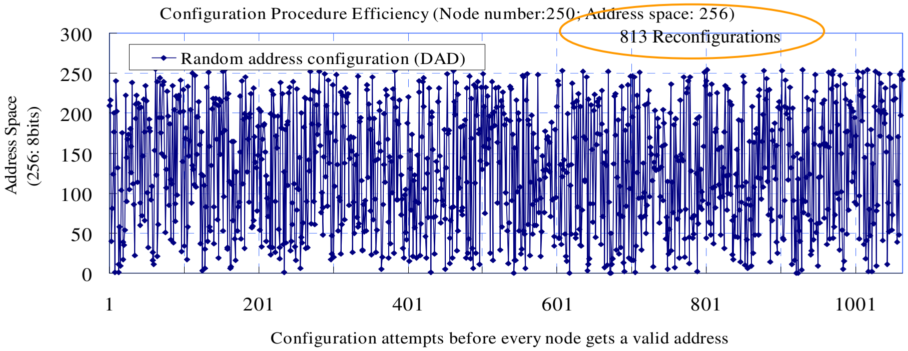
Fig.3. shows the configuration procedure efficiency that is measured by the total number of configuration attempts during the configuration. Here a configuration attempt refers to a round of address request and reply. As shown in the simulation results in Fig.3. (a) and (b) , to configure addresses for 100 nodes in an address space of 128, random address configuration (DAD) requires 205% configuration attempts of that of SDIC. And to configure addresses for 250 nodes in an address space of 256, DAD requires about 425.2% configuration attempts of that of SDIC, as shown in Fig.3. (c) and (d). DAD requires much more attempts and has many reconfigurations to have each sensor node a valid address. This is because that the probability of address conflicts is high for configuring a number of addresses in a small address space by the random address selection. In contrast, SDIC has a high efficiency for each sensor node to successfully get a valid address in each configuration attempt. This is because that SDIC utilizes a temporarily centralized and deterministic operation of address servers, leading to few address conflicts.


Configuration Procedure Efficiency (Node number:100; Address space: 128)

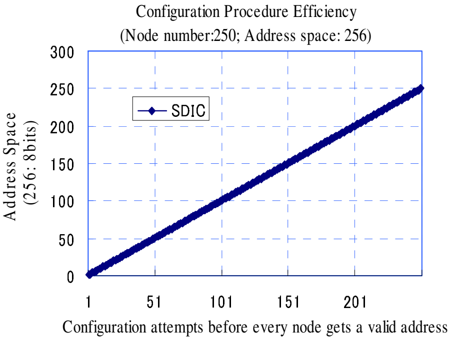
(a) Performance of DAD approach for configuring 100 nodes in address space of 128



(b) Performance of SDIC approach for configuring 100 nodes in address space of 128



(c) DAD approach for configuring 250 nodes in address space of 256



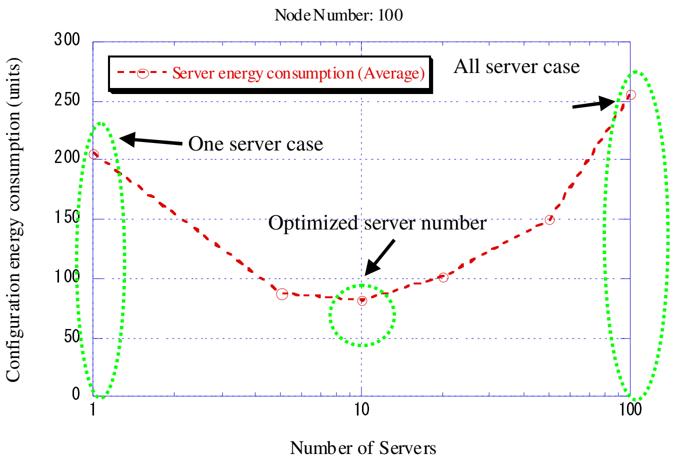
(d) SDIC approach for configuring 250 nodes in address space of 256

Fig. 3. Configuration procedure efficiency. To configure addresses for 100 nodes in an address space of 128, random address configuration (DAD) requires 205% configuration attempts of that of SDIC. And to configure addresses for 250 nodes in an address space of 256, DAD requires about 425.2% configuration attempts of that of SDIC.

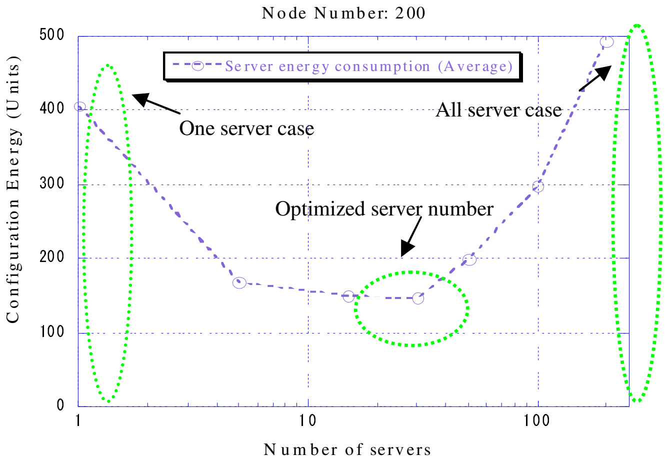
B. Configuration Scalability: The Impact of Server Term Control

Fig.4 shows the impact of server number during the configuration on the average energy consumption at the server nodes. The energy is measured by the energy unit that is the energy consumption in transmitting or receiving a configuration message. Since every server has the same server term, the number of servers reveals the server term of a server. More number of servers means small server term of an address server. The energy consumed at the server nodes is large in cases of either the server number is very small or the server number is very large. When there is only one server in the configuration, much configuration messages, such as AREQ and AREP, overhead will focus on the server, leading to large energy consumption. Further, there is a much unbalance of energy consumption between the address server and clients. On the other hand, when the entire nodes are selected to perform a server's role the energy consumption is large at both server and client. This is because that many servers causes much server rotation operations, leading to the large energy consumption.

One the other hand, we can observe that there is a optimized number of servers, which achieves the smallest energy consumption at the server nodes. When the server number approaches to a certain number such as 10 in the network of 100 nodes, and 15 of a network with 200 nodes, the energy consumption of servers has the lowest level. This result is in accordance with the analysis of the optimized server number.



(a) The impact of the server number (100 nodes case)



(b) The impact of the server number (200 nodes case)

Fig. 4. Impact of the server term control. There exist optimized server numbers, which result in the smallest energy consumption at server nodes.

7 Conclusion

This paper introduced a SDIC approach that self-organizes the sensor nodes like the flight formation of birds to cooperatively configure energy efficient addresses. SDIC configures addresses for a large number of sensor nodes within a small address space by constructing a controllable and optimized network infrastructure. Like the role management among birds in flight formation, sensor nodes are self-organized with the roles of address servers and address clients to manage the use of address space. SDIC controls server term for the server's rotation among sensors to achieve server robustness and the energy efficiency of configuration. Consequently, the roles and duties of server/clients are dynamically assigned to sensor nodes and are performed in a distributed way.

SDIC configures local unique addresses sequentially, leading to small address size. The configuration is scalable with regard to address conflicts, energy consumption of servers, and information maintained in the servers. Simulation results show that SDIC avoids address conflicts and requires little reconfiguration compared with the conventional address autoconfiguration schemes, especially when there are many nodes in the network. With the efficient server term control, SDIC achieves low and balanced energy consumption among sensor nodes. The use of energy-efficent addresses in sensor network and the bird-flight formation inspired self-organization also highlight the future applications of sensor networks, in which sensor nodes can perform various sensing tasks by cooperatively sharing the roles of network managements with energy efficiency and scalability.

References

1. Estrin, D., Govindan, R.: Next Century Challenges: Scalable Coordination in Sensor Networks. In: ACM/IEEE Conf. MobiCom 1999, pp. 263–270 (1999)
2. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications, 660–670 (October 2002)
3. Mainwaring, A., Polastre, J., Robert, S., David, C., John, A.: Wireless Sensor Networks for Habitat Monitoring. In: ACM International Workshop on Wireless Sensor Networks and Applications, pp. 88–97 (2002)
4. Nesargi, S., Prakash, R.: MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network. In: INFOCOM 2002, pp. 1059–1068 (2002)
5. Vaidya, N.H.: Weak duplicate address detection in mobile ad hoc networks. In: Proc. of ACM MobiHoc, pp. 206–216 (2002)
6. Perkins, C.E., Malinen, J.T., Wakikawa, R., Belding-Royer, E.M., Sun, Y.: IP Address Autoconfiguration for Ad Hoc Networks. Internet Engineering Task Force, MANET Working Group (July 2000), draft-ietf-manet-autoconf-01.txt
7. Gunes, M., Reibel, J.: An IP Address Configuration Alorgrithms for Zeroconfig Mobile Ad Hoc Networks. In: International Workshop on Broadband Wireless Ad Hoc Networks and Services (2002)
8. Aboba, B., Cheshire, S.: Dynamic Host Configuration Protocol (DHCP) Domain Search Option. RFC 3397 (November 2002)
9. Elson, J., Estrin, D.: Random, Ephemeral Transaction Identifiers in Dynamic Sensor Networks. In: Int'l Conf. Distributed Computing Systems (ICDCS 2001), pp. 459–468 (2001)
10. PalChaudhuri, S., Du, S., Saha, A., Johnson, D.: Treecast: A Stateless Addressing and Routing Architecture for Sensor Networks. In: 4th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN 2004), pp. 221–228 (2004)
11. Schurgers, C., Kulkarni, G., Srivastava, M.B.: Distributed assignment of encoded mac addresses in senosr networks. In: Proc. of ACM Mobihoc, pp. 295–298 (2001)
12. Motegi, S., Yoshihara, K., Horiuchi, H.: Address Autoconfiguration for Event-Driven Sensor Network. IEICE Transaction on Communication, 950–957 (March 2005)
13. Zhou, H., Ni, L.M., Mutka, M.W.: Prophet address allocation for large scale MANETs. In: Proc. INFOCOM, pp. 1304–1311 (2003)
14. The IETF Zero Configuration Networking (zeroconf) Working Group, `http://www.zeroconf.org/zeroconf-charter.html`
15. Nahavandi, A.: Art and Science of Leadership. Prentice Hall, Upper Saddle River (2000)
16. Teng, R., Morikawa, H., Aoyama, T., Zhang, B.: Network-Layer and MAC-layer address autoconfiguration in self-organized sensor networks. In: 6th International Conference on ITS Telecommunications, pp. 1005–1010 (June 2006)
17. Yamazaki, K., Sezaki, K.: Spatio-temporal addressing scheme for mobile ad hoc networks. In: TENCON 2004, pp. 223–226 (2004)
18. Weimerskirch, H., Martin, J., Clerquin, Y., Alexandre, P., Jiraskova, S.: Energy saving in flight formation. Nature (October 18, 2001)