www.ietdl.org

Published in IET Wireless Sensor Systems
Received on 29th January 2011
Revised on 9th July 2011
doi: 10.1049/iet-wss.2011.0022

Special Issue: Wireless Sensor Networks 2011

ISSN 2043-6386

Designing a self-organised system for service substitution and placement in wireless sensor networks

S.A. Asghari[1] *H.G. Zadeh*[2] *H. Taheri*[2] *H. Pedram*[1]

[1]*Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran*
[2]*Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran*
E-mail: seyyed_asghari@aut.ac.ir

Abstract: Since wireless sensor networks (WSN) as *ad hoc* networks do not have infrastructures and their topologies change dynamically, network nodes should be able to take suitable decisions without needing to know the network topology. One of these decisions is selecting the appropriate policy for service distribution of the whole network. The method of service distribution in the network is uniform by default. Therefore there may be some nodes in the network that provide service and there is less or no demand for them. So, based on the default attitude, energy and power consumption are high and since WSN are sensitive to the energy and power consumption, this is undesirable. In this study, a self-organised system for service placement and substitution in WSN is presented. By utilising this method, nodes can act automatically in delivering a service. Therefore the nodes that have no service request will be automatically omitted in the network. By utilising the proposed method of this study, the energy and power consumption of a wireless sensor network can be reduced by more than a half.

1 Introduction

Wireless networks, and especially wireless sensor networks (WSN), have been the field of many researches in recent years because of the increase in their application.

Since WSN are without infrastructures and network topologies change dynamically, the nodes should be able to take suitable decision without knowing the entire topology of the network. Self-organisation is a solution for the mentioned problems and other problems in WSN that have been analysed in many papers.

The self-organisation topic progresses so fast and is therefore one of the most important challenges in the entire distributed computer systems [1–4].

Self-organisation is not engineering creativity, but it is a pervasive content in many of the biological, physical, social and economical systems.

Many definitions are presented for self-organisation. However, the definition of [5] is related to the present paper, that is, 'Self-organisation is a process in which the pattern at the global level of a system emerges solely from numerous interactions among the lower level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern'.

One of the characteristics of self-organised systems is that they do not have centralised control, scalability and positive/negative feedbacks [1–4, 6].

From centralised control systems to distributed systems and self-organised systems, the scalability is increased and the deterministic feature is decreased (Fig. 1).

Many efforts have been made to gain inspiration from self-organisation in biological systems to recommend methods for computer systems, but such a method transfer, that is, from biological systems to computer systems has many limitations and is somehow impossible. Therefore it is needed to design methods that are independent from biological systems [7–9].

In this paper, the field of self-organisation, which has been less dealt with so far, is analysed. In fact, most of the works in the self-organisation field are related to clustering and routing, and so service distribution methods in the network and how to access them have been less analysed.

Section 2 of this paper presents the review of literature in the self-organisation field. Section 3 presents the explanation of service distribution algorithm. Section 4 presents the proposed method for solving this problem. Experimental results are presented in Section 5.

2 Review of the literature

From the point of view of open system interconnection (OSI) reference model, clustering and service distribution are implemented on an application layer. In the clustering field, several algorithms are presented [10, 11].

In [12], a brief study on the self-organisation field in *ad hoc* networks and wireless sensors has been done. However, it is clear from this study that the author introduces self-organisation in clustering and has not considered the service distribution method in such networks. In fact, what is under question in service distribution or service placement [13] in WSN is the accurate decision making of the server node without having a comprehensive perception of the network

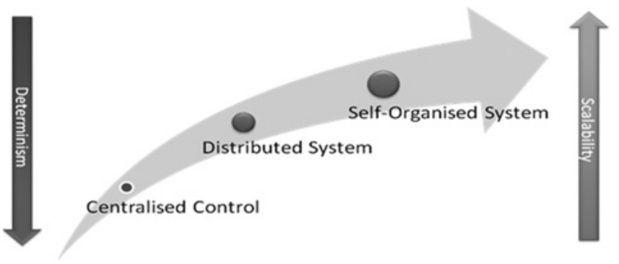

Fig. 1 *Determinism and scalability comparison*

topology for requests produced for that service. This accurate decision making can consist of migration to the other nodes, producing a replicated copy and/or omitting the related service.

Utilising substitution algorithm is not a new subject. Server substitution algorithms in internet medium can be considered as one of the examples. Many methods have been delivered in the server substitution field and web proxies for performance optimisation of some of the parameters [14–17].

The utilised algorithms in these references need to be aware of the network topology. The proposed algorithm for service replication in internet based on service overlay networks (SON) can also be a similar field [18].

The problem associated with this algorithm is its centralisation and needing to know all the source/destination pairs in SON.

The other field of this subject is grid computing. In the proposed structure of the [19], servers are replicated to be in correspondence with users' demands. This is similar to the idea considered in WSN in this paper. However, its problem is because of the centralised algorithm and that service management should consider the place of the desired service (centralised control).

In [20], several substitution algorithms are presented in a grid medium. Not all of them, however, support service migration.

Owing to the problems that have been considered for the mentioned algorithms, they cannot be used as the self-organised solution for service distribution in WSN. As it was mentioned before, scalability, not having the centralised control and using local information are the characteristics needed for the suitable response to the dynamic changes in WSN to prepare suitable servicing for users.

One of the few studies that accurately investigate service distribution in WSN is [6]. The differences of the infrastructure of this study are considered to be as follows: (i) The medium is wireless and its bit error rate is higher than that of the wired medium. (ii) There is the problem of bandwidth shortage in wireless networks, but it is assumed that the network nodes used for servicing are connected to an infinite source of energy.

The next section presents the proposed algorithm in [6] and also a method for solving the mentioned problems of this algorithm.

3 Service substitution algorithm

Ad hoc service grid (ASG) [21] structure is considered for implementation. This structure is a network of multi-purpose nodes of service cube in which the service is injected.

These nodes have a wireless transceiver and can communicate with each other by means of wireless communication. Each node's coverage varies from one node to several nodes (based on type of node transceivers). ASG structure is very flexible and scalable and network nodes can be added to or removed from them.

Other nodes of this network are clients (e.g., cell phones and personal digital assistant (PDA)) that are very dynamic. Fig. 2 shows a part of the ASG network in which *R* represents a service sample in a node, black triangles represent users that are connected to a node and LS represents the cluster head nodes that searches for service.

Communication links between nodes (cubes) are wireless and services like *R* are distributed in the network according to a specific pattern based on the service demand rate. The communication between the user (like cell phones, PDA etc.) and each network node (usually the closest node to the user) is wireless (the communication between the user and network nodes can be wired too).

Since the control of services and their suitable methods of distribution are the responsibilities of the application layer, it is assumed in this paper that there exists the suitable software which is installed on network nodes [22, 23].

Therefore the algorithm that is implemented on the theme is presented here. In the ASG scheme, which is a distributed structure, services are not necessarily presented equally, because the presentations of these services are based on demands. Now, by using the demands and service distribution statistical information, network behaviour parameters can be adjusted. For example, in the wireless sensor network, in which their performance and power consumption are the main matters of concern, the nodes with fewer requests can be omitted.

For responding to the received request and for less sending of the nodes, the existed service in each node can encounter with the following attitudes [13]:

Migration: The desired node under some conditions moves to the node that is closer to the requests.
Replication: A replica similar to the main replica is produced by the node that has the service and is sent to the destination node.
Omitting: If the received request rate of a specific service is fewer than the threshold limit in the determined time, this service will be omitted.

What is important in this paper is the migration regularity. The migration of a sample service is more important than the other two costs (replication and omitting) since if the desired

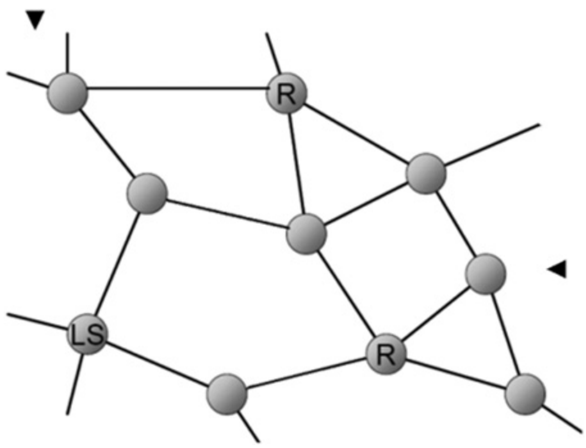

Fig. 2 *Part of ASG*

service moves from one node to the other nodes, it is closed to some requests and goes away from other ones.

As a request, migration should be such that finally the numbers of the requests sent are fewer than in the previous state. Therefore two following conditions should be satisfied

1. $$M_v(r, u, t_0, T_h) \succ \sum_{wi \in Nv\backslash\{u\}} M_v(r, w_i, t_0, T_h)$$

2. $$\frac{\Delta t}{T_h} \sum_{i=0}^{(T_h/\Delta t)-1} b_v(r, u, t_0 - i\Delta t, I) \succ \eta$$

And in condition 2, we have

3. $$b_v(r, u, t_0 - i\Delta t, I) = \begin{cases} 1, & M_v(r, u, t_0, T_h) \succ \sum_{wi \in Nv\backslash\{u\}} M_v(r, w_i, t_0, T_h) \\ 0, & ow \end{cases}$$

In condition 1, $M_v(r, u, t_0, T_h)$ shows the number of requests that node v receives in the time of T_h for the sample service of r from the neighbour node of u. $M_v(r, w_i, t_0, T_h)$ has the same concept but is related to the received requests from node w_i. t_0 is also the time index and shows the time in which the migration regulatory is analysed. N_v is the collection of node v neighbours. If condition 1 is satisfied, it means that the number of received requests from u is more than all of the received requests from other neighbours of v and if condition 2 is satisfied, the migration will happen. However, a situation may occur in which the requests from two neighbours of two different directions differ slightly with each other. For instance, randomly and time variably, the number of received bytes from u and w neighbours differ slightly from each other. In this case, for example, r migrates to node w randomly; however, if it stayed in that node, it would have been better. For preventing it, T_h, is divided into very small time slots of Δt (T_h mod $\Delta t = 0$) and condition 1 is analysed in all these slots and if condition 3 equals 1 in η percent of these slots, migration takes place.

4 Proposed method for sample service behaviour by the assumption of a finite source of energy

As it is explained in [21], cubes are connected to a source of energy. In fact, this example maps a medium (a shopping centre), in which there is no problem of providing energy. However, in many military or relief effort applications, providing such a source of energy is not easy.

If the proposed method of [6] is considered for the finite energy state, the following problems may arise: If the node that has the service replica is turned off because of the battery discharging, the service replica goes out of the access area, the users who are receiving the service waste their energy because of disconnection and an unwanted traffic occurs because of the repeated request sending. For solving such problems, Fig. 2 algorithm is presented. This figure presents all the actions that may take place.

At first, it should be analysed whether the received requests have been sufficient for the service or not. If it is not sufficient, the service replica of the node is omitted and the node's turning off is not important.

```
begin
 if  Idle(r)  then  r. Dissolve()
 else
  if  β < threshold  then  V= GetMoveTarget(r)
  else
   (v,w) = GetReplicationTargets(r)
   if   v≠ v'  and w≠ v'  then r.Replicate(v,w)
   else
     v = GetMigrationTarget (r)
     if  v≠ v'  then  r. Migrate(v)
     endif
   endif
  endif
 end

procedure  GetMoveTarget(r)
u and w ϵ N_v
begin
  if  M_v(r,u,t_0,Th) ≻ Σ_{wi∈Nv\{u}} M_v(r,w_i,t_0,T_h)
 then return  u
  else  {w∈N_v | ∀w_i∈N_v: M_v(w) ≥ M_v(w_i)}
  then return  w
endif
end
```

Fig. 3 *GetReplicationTarget (r) and GetMoveTarget (r) procedures*

In the next step, the battery amount of node (β) is analysed and if it is less than the threshold amount, the GetMoveTarget (v) procedure is performed.

In this step, it is tried to not to lose the service replica. Therefore condition 1 is analysed first and if a node with this condition is not found, the following condition is analysed:

4. $$\{w \in N_v | \forall w_i \in N_v: M_v(w) \geq M_v(w_i)\}$$

Condition 4 means that if node u is not found in the first condition, the neighbour, which has more requests, is selected for migration. However, question arises as to how the users and service senders' nodes should be informed about the new place of service. Since the battery of node v has not been discharged yet, and the requests are sent to v, node v receives the requests and sends them to the new place of service replica by using the proposed method in [24]. Also owing to the fact that the service searching system is situated in the cluster head, this change is informed to the cluster head.

After that the response is sent back to the source by the new node that has the service, service senders diagnose the new place of service by the source addresses of the packets.

GetMoveTarget(r) and GetReplicationTarget(r) procedures (Fig. 3) are presented in [6]. In the following procedures, v' shows the invalid node (a procedure may turn back a node that satisfies the related condition but that node has not existed).

5 Experimental results

In order to simulate, 36 wireless nodes have been considered. The infrastructure of the desired simulation is a distributed wireless network.

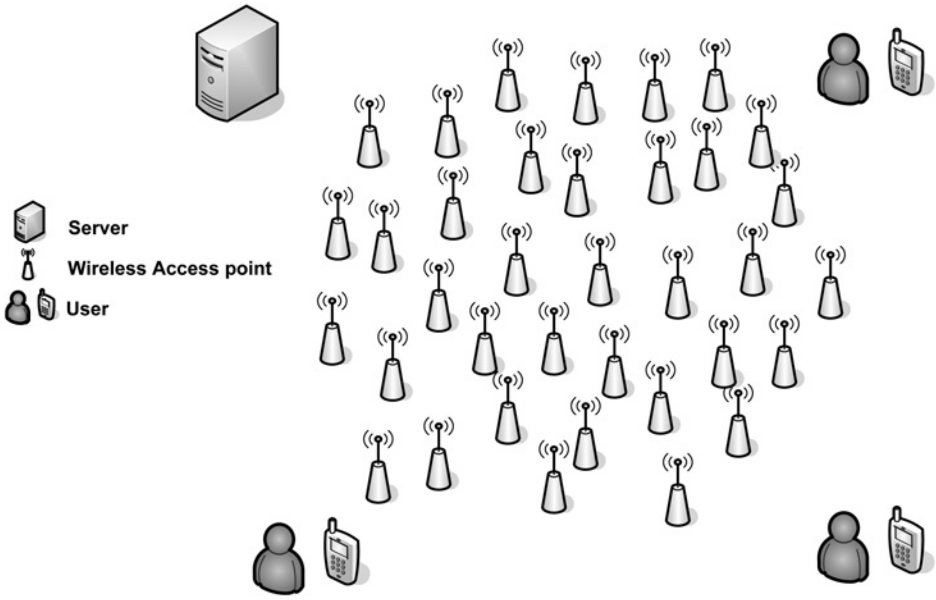

Fig. 4 *Network infrastructure for implementation*

Network cubes receive their needed information from the main server of the network and users receive their needed information from the closest cube around them.

The desired infrastructure is shown in Fig. 4.

The average number of hops between nodes is 7.1 hops. This test is increased from fewer to more number of replicas. By increasing hop numbers that can be called replication radius, the number of the needed replication is reduced (Fig. 5).

The results of this test, from the point of view of energy consumption as a result of increase in cube number, are shown in Fig. 6.

The length of transferred messages between nodes is considered to be 1024 bits (1 kb). For each bit, 50 nJ energy is consumed. The initial energy required for the network set-up is 0.5 J.

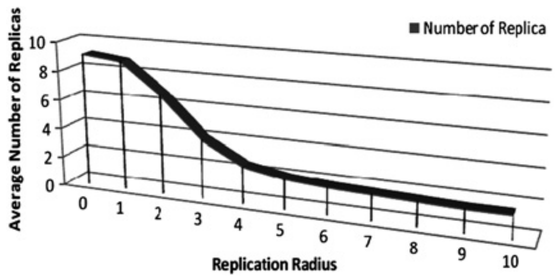

Fig. 5 *Number of replications for replication radius increasing*

The desired experiment for analysing the proposed method was also carried out by more nodes. In this experiment, 100 cubes in a wireless sensor network were considered (Fig. 7).

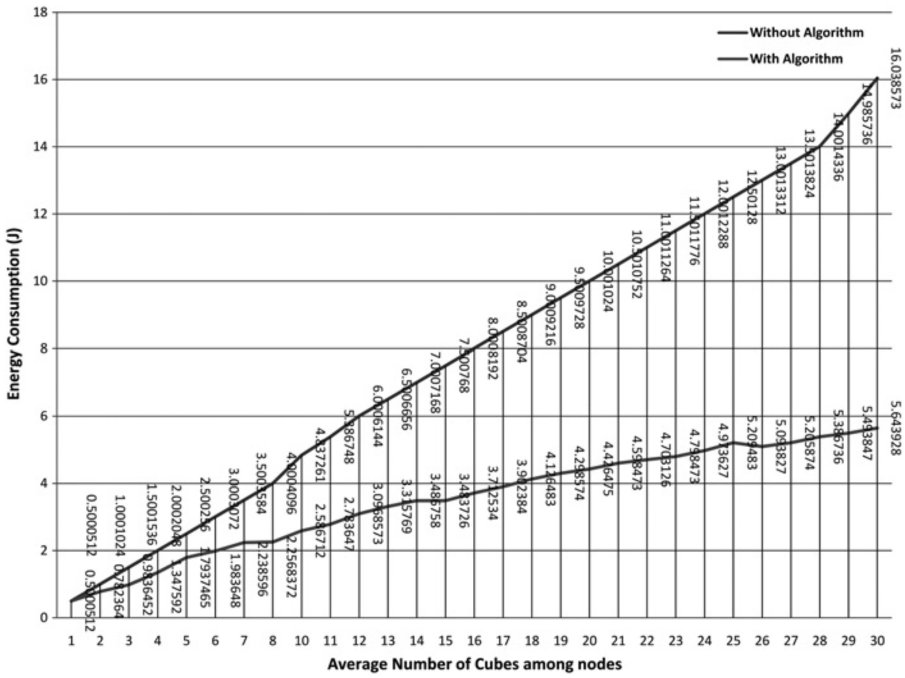

Fig. 6 *Energy consumption for cube number increasing up to 36 cubes*

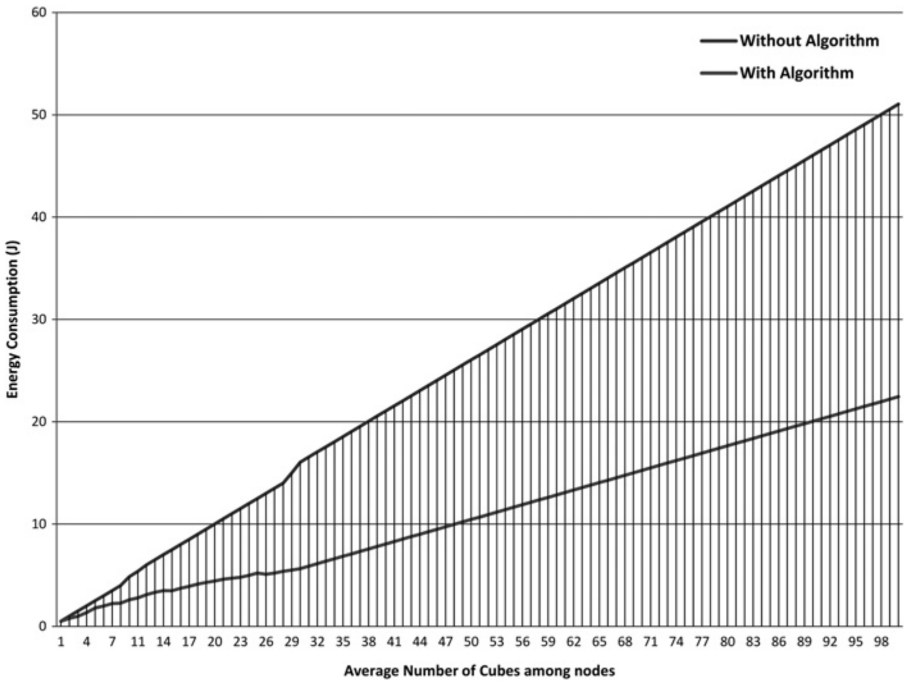

Fig. 7 *Energy consumption for cube number increasing up to 100 cubes*

Without using the proposed method, the energy consumption of the whole network was 51.03857 J, whereas by utilising the proposed method, this energy consumption equalled 22.44393 J.

Therefore by using the proposed method, the energy and power consumption in the network can be reduced by more than a half. This reduction was performed by utilising the proposed method and bringing the cubes, such that their demand rates were less than the specified threshold, to the sleep mode.

6 Conclusions

Many works have already been done on self-organised networks. However, service substitution and placement of these networks need optimisation. In this paper, an attitude is presented for automatic service organisation in a wireless sensor network that can be utilised in other network infrastructures such as WSN. Energy and power consumption of these networks are the main matters of concern.

The presented method can reduce the power consumption significantly, because the cubes that do not provide any service, and for which there is no demand, can omit themselves from the network automatically.

Therefore by utilising the proposed method, the cubes that have less demand rates than the specified threshold are automatically omitted or go to the sleep mode. It is expected that by using this method, energy and consumption power are reduced to a great extent and the results of this paper prove it. The method proposed in this paper can reduce the energy and power consumption of the wireless sensor network by more than a half.

7 References

1 Kephart, J., Chess, D.: 'The vision of autonomic computing', *IEEE Comput.*, 2003, **36**, (1), pp. 41–50

2 Horn, P.: 'Autonimic computing: IBM's perspective on the state of information technology', http://www.research.ibm.com autonomic/manifesto/autonomic_computing.pdf, October 2001 (accessed 8 May 2006)

3 Vakali, A., Pallis, G.: 'Content delivery networks: status and trends', *IEEE Internet Comput.*, 2003, **7**, (6), pp. 68–74

4 Androutsellis-Theotokis, S., Spinellis, D.: 'A survey of peer-to-peer content distribution technologies', *ACM Comput. Surv.*, 2004, **36**, (4), pp. 335–371

5 Camazine, S., Deneubourg, J., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E.: 'Self-organization in biological systems' (Princeton University Press, 2001)

6 Herrmann, K.: 'Self-organized service placement in ambient intelligence environments', *ACM Trans. Auton. Adap. Syst.*, 2010, **5**, (2), Article 6, pp. 1–39

7 Ohtaki, Y., Wakamiya, N., Murata, M., Imase, M.: 'Scalable and efficient ant-based routing algorithm for Ad-Hoc networks', *IEICE Trans. Commun. E Ser. B*, 2006, **89**, (4), pp. 1231–1238

8 Ong, S.L., Lai, W.-K., Tai, T.S.Y., Hoe, K.M., Ooi, C.H.: 'Application of ant-based template matching for web documents categorization', *Informatica*, 2005, **29**, (2), pp. 173–182

9 Ramos, V., Merelo, J.J.: 'Self-organized stigmergic document maps: environment as a mechanism for context learning'. AEB'2002 – First Spanish Conf. on Evolutionary and Bio-Inspired Algorithms, February 2002, pp. 284–293

10 Rabiner Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: 'Energy-E_cient communication protocol for wireless microsensor networks'. 33rd Hawaii Int. Conf. on System Sciences, 2000

11 Basagni, S.: 'Distributed clustering for ad hoc networks'. Proc. IEEE Int. Symp. on Parallel Architectures, Algorithms, and Networks (I-SPAN), June 1999, pp. 310–315

12 Dressler, F.: 'A study of self-organization mechanisms in ad hoc and sensor networks', *Elsevier Comput. Commun.*, 2008, **31**, (13), pp. 3018–3029

13 Herrmann, K.: 'Self-organizing replica placement – a case study on emergence'. First Int. on Conf. Self-Adaptive and Self-Organizing Systems, (SASO 2007), 2007

14 Jia, X., Li, D., Hu, X., Du, D.: 'Optimal placement of web proxies for replicated web servers in the internet', *Comput. J.*, 2001, **44**, (5), pp. 329–339

15 Jia, X., Li, D., Hu, X., Du, D.: 'Placement of read-write web proxies in the internet'. Proc. 21st Int. Conf. on Distributed Computing Systems (ICDCS 2001), April 2001, pp. 687–690

16 Li, K., Shen, H.: 'Optimal placement of web proxies for tree networks'. Proc. 2004 IEEE Int. Conf. on e-Technology, e-Commerce and e-Service (EEE'04), 2004, pp. 479–486

17 Tenzakhti, F., Day, K., Ould-Khaoua, M., Obadiat, M.S.: 'Placement of web proxies with server capacity constraints'. Proc. High Perform. Comput. Symp. (HPC 2004), April 2004, pp. 101–106

doi: 10.1049/iet-wss.2011.0022

18 Liu, K.Y., Lui, J.C., Zhang, Z.-L.: 'Distributed algorithm for service replication in service overlay network'. Proc. Third IEEE/IFIP TC-6 Networking Conf., (Networking 2004), 2004

19 Lee, B.-D., Weissman, J.B.: 'Dynamic replica management in the service grid'. Proc. Tenth IEEE Int. Symp. on High Performance Distributed Computing, (HPDC-10'01), August 2001

20 Andrzejak, A., Graupner, S., Kotov, V., Trinks, H.: 'Algorithms for self-organization and adaptive service placement in dynamic distributed systems'. Technical report HPL-2002-259, HP Laboratories, Palo Alto, 2002

21 Herrmann, K., Geihs, K., M̈uhl, G.: 'Ad hoc service grid – a self-organizing infrastructure for mobile commerce'. Proc. IFIP TC8 Work. Conf. on Mobile Information Systems (MOBIS2004), IFIP – International Federation for Information Processing, September 2004

22 Hellenschmidt, M., Kirste, T.: 'SodaPop: a software infrastructure supporting self-organization in intelligent environments'. Proc. Second IEEE Conf. on Industrial Informatics, INDIN 04, June 2004

23 Herrmann, K.: 'MESHMdl – a middleware for self-organization in Ad hoc networks'. Proc. First Int. Workshop on Mobile Distributed Computing, (MDC'03), May 2003

24 Herrmann, K., M̈uhl, G., Jaeger, M.A.: 'A self-organizing lookup service for dynamic ambient services'. 25th Int. Conf. on Distributed Computing Systems, (ICDCS 2005), Piscataway, NJ, USA, June 2005