

# Scalable Service Migration in General Topologies

Konstantinos Oikonomou  
Dept. of Informatics  
Ionian University, Corfu, Greece  
okon@ionio.gr

Ioannis Stavrakakis and Alexios Xydias  
Dept. of Informatics & Telecomms  
University of Athens, Athens, Greece  
{ioannis, std02080}@di.uoa.gr

## Abstract

*Service placement has typically been studied through the formulation and solution of a 1-median problem that is known to be complex and require global information. As such approaches are clearly not scalable in dynamic and large-scale environments, a more reasonable approach to service placement for large, ad hoc and autonomic environments would be through service migration. Such a migration policy has recently been proposed for tree topologies. The focus of the present paper is to devise effective service migration policies for general topologies with equal weight links. The proposed strategy – that incorporates a 2-phase polling mechanism of local scope and, consequently, induces in general a certain overhead - is shown to outperform the previous one (i.e., getting closer to the optimal placement) for such general topologies. Finally, a hybrid migration policy (combining the present and the past ones) is considered and shown through simulations to be capable of reducing significantly the overhead effects of the aforementioned polling mechanism.*

## 1 Introduction

The service placement problem is encountered in various networks such as transportation networks, supply networks and communication networks, [7], [13]. The globalization of the Internet and the proliferation of services and service demands have necessitated the careful selection of the location of the service. The goal is to bring the service provision points (referred to here as *service nodes*) close to the demand in order to minimize communication resource consumption and enhance the Quality of Service (QoS) of the provided service. Due to the recent technological changes (e.g., the introduction of powerful machines, the service proliferation, the generalization of high-power computing) the traditional problem of placing relatively *few* big services in one of the *few* (powerful) potential service provider facilities (big network elements) is increasingly being trans-

formed into a problem of placing *numerous* services to one of the *numerous* potential service providers (network elements and possibly service producers).

This paper focuses on the problem of determining the optimal service placement (that is, determining the optimal position of the service node), so that some average cost associated with the provision of this service is minimized. When  $p$  ( $p \geq 1$ ) service nodes are considered, the optimal placement can be determined by formulating and solving a *p-median problem*, [7]. The *p-median problem* has been shown to be an *NP-hard* problem for general graphs, [3]. Various attempts, [11], [4], [5], [12], [1], have been proposed in the past, mostly aiming to achieve suitable approximation solutions. More recent work in the area can be found in [6], [9]. In most cases – apart from the heuristic or approximation policies – the determination of the optimal service node requires some *global network information* (e.g., the network topology and the service demands of all network nodes). Since such (traditional) approaches are complex and are based on global information availability, they are non-scalable for networking environments supporting numerous services, service users and network elements, which are also expected to be fairly dynamic.

A more reasonable approach to service placement for large, ad hoc and autonomic environments would be through *service migration*, [10]. That is, consider policies for moving the service position (one hop/node at a time) towards more effective positions based on *local information*, instead of requiring global information and solving continuously (in response to dynamic changes) a large optimization problem, [10], [8], [13], [14]. As it was shown for the migration policy introduced in [10], the service node needs to simply monitor the *aggregate* amount of data associated with the particular service that are exchanged through its neighbor nodes and decide on the service movement based *exclusively* on the information gathered through the (local) monitoring process. This information was shown to be *adequate* in order for the service to move towards and till the end of a monotonically cost decreasing path in topologies like trees with a unique *shortest path tree* (SPT), [2], [10].

The *service movement criterion* presented in [10], to be referred to hereafter as *Criterion A*, is sometimes not able to move the service until the *end* of a cost decreasing path, particularly when *alternative* shortest paths can be utilized after a service movement to a new neighbor node (i.e., more than one SPTs in the network). Eventually, *Criterion A* does not allow for further cost reduction, even though this would be achievable by a single movement of the service to a neighbor node. This particular inefficiency motivates the introduction of another service migration criterion, i.e. *Criterion B*, presented here, that is shown to be capable of moving the service along and till the end of a monotonically cost decreasing path without the unique SPT requirement (note that in general, the end of a monotonically cost decreasing path may not be the *optimal position* – i.e., the node at which the service cost is minimized). *Criterion B*, however, depends on non-local information in the general case, and therefore, general topologies with equal link weights are also considered in this paper that allow for further refinement of *Criterion B*, thus resulting in *Criterion B<sub>e</sub>*, that is shown to depend exclusively on local information.

*Criterion B<sub>e</sub>* is based on a *2-phase polling mechanism* which requires two *tentative* service movements before a decision for a service movement to a certain neighbor node is finalized. Therefore, a hybrid migration policy that combines and utilizes both criteria is proposed, which significantly reduces the introduced overhead due to the polling mechanism, as shown through simulations.

In Section 2 the network environment is described along with the problem formulation and some useful definitions. In this section the applicability of *Criterion A*, proposed in [10], and its limitations are also presented motivating the introduction of *Criterion B*, presented in Section 3. The particular case of a general network topology of equal link weights is considered in Section 4 where *Criterion B<sub>e</sub>* is introduced and the conclusions are drawn in Section 5.

## 2 Problem Formulation and Criterion A

The network topology is represented by an undirected graph  $G(V, E)$ , where  $V$  is the set of nodes and  $E$  the set of links among them. Let  $S_v$  denote the set of nodes that have a direct link with node  $v$ . Let  $(u, v)$  denote a certain link of the graph between node  $u$  and node  $v$ . Let, also, each link be assigned a positive integer referred to as *weight*. Let  $d_{u,v}$  denote the *distance* for the particular  $G(V, E)$  graph between node  $u$  and node  $v$ , corresponding to the summation of the weights along a *shortest path* among both nodes (for the same node  $v$ ,  $d_{v,v} = 0$ ). Alternatively,  $d_{u,v}$  may be referred to as the *traveling cost* between nodes  $u$  and  $v$ . Let  $\lambda_v$  denote the rate at which data packets associated with a particular service are transferred through the network

(along a shortest path) between node  $v$  and the service node:  $\lambda_v$  will be referred to as the *service demands* of node  $v$ .

For a given *service node* location (normally node  $y$  in this paper), it is the role of the employed *routing protocol* to identify a suitable route throughout the network links and nodes towards the particular destination (i.e., the service node). In this work it is assumed that the routing protocol is capable of forwarding the nodes' *data packets* (corresponding to their service demands) over a *shortest path* towards their final destination, [2]. Eventually, a *shortest path tree* (SPT), [2], is created rooted at the particular service node. Let  $SPT^{(x)}$  denote the particular SPT when the root is node  $x$ . If the root of the SPT is the service node  $y$ , then all data packets corresponding to the particular service are forwarded over the branches of  $SPT^{(y)}$  towards the service node  $y$ .

Let  $C_y^{(x)}$  denote the total *cost* for all network nodes using a particular service when the service is located at node  $y$  and the data packets corresponding to the service demands are forwarded in the network along the branches of  $SPT^{(x)}$ , for some node  $x$ . Let also  $d_{y,v}^{(x)}$  denote the distance between node  $y$  and node  $v$  in  $SPT^{(x)}$  (note that  $d_{y,v}$  corresponds to the distance between node  $y$  and node  $v$  over a shortest path in graph  $G(V, E)$ ). Eventually,

$$C_y^{(x)} = \sum_{v \in V} \lambda_v d_{y,v}^{(x)}. \quad (1)$$

Note that  $C_y^{(x)} \geq C_y^{(y)}$ , since  $SPT^{(y)}$  by definition provides the routes to a service located at node  $y$  (its root), of a cost that cannot be higher than that of the roots dictated by any other  $SPT^{(x)}$ , for  $x \neq y$ .

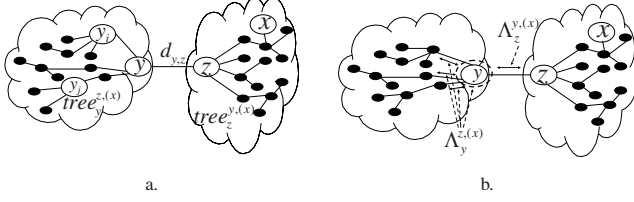
### 2.1 Aggregate Service Demands

The objective in this paper is to move the service to neighbor nodes in an attempt to reduce the aforementioned cost and stop *service migration* when no further cost reduction is achievable. Let “service movement  $y \rightarrow z$ ” denote the movement of the service from node  $y$  to some neighbor node  $z$ ,  $z \in S_y$ , assuming to be completed in one *time unit*.

Suppose that a particular link  $(y, z)$  is removed from  $SPT^{(x)}$ . Let the set of nodes of the subtree of  $SPT^{(x)}$  that includes node  $y$  ( $z$ ) be denoted as  $tree_z^{y,(x)}$  ( $tree_z^{z,(x)}$ ), respectively (see Figure 1.a). Obviously,

$$V = tree_z^{z,(x)} \cup tree_z^{y,(x)}. \quad (2)$$

The determination of subtree  $tree_z^{y,(x)}$  of some  $SPT^{(x)}$  facilitates the description of the aggregate service demands generated from nodes  $z$  and below in  $tree_z^{y,(x)}$ . Assuming that the service is located at node  $y$ , data packets exchanged between the service node  $y$  and any node  $v \in tree_z^{y,(x)}$ ,



**Figure 1. Example for sets  $tree_y^{z,(x)}$  and  $tree_z^{y,(x)}$  and the corresponded  $\Lambda_y^{z,(x)}$  and  $\Lambda_z^{y,(x)}$  service demands.**

will be forwarded through node  $z$ . Eventually,  $z$  is the node through which the aggregate amount of data for all nodes  $v \in tree_z^{y,(x)}$ , are forwarded towards their destinations. The aggregate service demands of those nodes belonging to  $tree_z^{y,(x)}$ , denoted by  $\Lambda_z^{y,(x)}$ , are given by,

$$\Lambda_z^{y,(x)} = \sum_{\forall v \in tree_z^{y,(x)}} \lambda_v. \quad (3)$$

An example for a service located at node  $y$  is depicted in Figure 1.b. The particular value of  $\Lambda_z^{y,(x)}$  is made available through an appropriate monitoring mechanism, [10] which is not in the scope of this paper to be described.

## 2.2 Criterion A

The service migration philosophy, briefly described in Section 1 and in more detail in [10], is based on per hop service movements over neighbor nodes. In [10], the following migration criterion was proposed.

*Criterion A:* Service movement  $y \rightarrow z$  takes place, if  $\Lambda_y^{z,(y)} < \Lambda_z^{z,(y)}$  is satisfied.

When the condition  $\Lambda_y^{z,(y)} < \Lambda_z^{z,(y)}$  is satisfied, it is ensured that  $C_z^{(y)} < C_y^{(y)}$ , [10]. Since both  $\Lambda_y^{z,(y)}$  and  $\Lambda_z^{z,(y)}$  can be made available locally at the service node  $y$  using an appropriate monitoring mechanism for the aggregate service demands, [10], Criterion A can be employed in large scale peer-to-peer and autonomic networking environments. As it was shown in [10], under Criterion A the service moves along a cost decreasing path.

One limitation of Criterion A is that it may not follow the cost decreasing path until the end. This is due to the fact that Criterion A considers  $SPT^{(y)}$  when considering a move to a node  $z$  ( $\Lambda_y^{z,(y)}$  and  $\Lambda_z^{z,(y)}$  correspond to the aggregate service demands forwarded over the branches of  $SPT^{(y)}$ ). Assume that Criterion A stops at some node  $y$ ; that is  $C_z^{(y)} \geq C_y^{(y)}$ ,  $\forall z \in S_y$ . If  $SPT^{(y)} \neq SPT^{(z)}$  though, as it may well be the case, then it may be that

$C_z^{(z)} < C_y^{(y)}$ , due to the utilization of shortest paths under  $SPT^{(z)}$  that are not part of  $SPT^{(y)}$ . Consequently, Criterion A may fail to discover a cost decreasing service movement  $y \rightarrow z$ .

Of course, if the network contains a unique SPT (e.g., in tree topologies), the service eventually arrives at the optimal position under Criterion A, [10]. A unique SPT exists, if  $SPT^{(x)}$  consists of the same set of links and nodes for all network nodes  $x \in V$ . In general topologies, for two different nodes  $y, z$ , it is possible that  $SPT^{(y)} \neq SPT^{(z)}$ , and therefore, multiple SPTs (i.e., more than one) exist in the network depending on the particular root.

## 3 Migration Criterion B

Assume that  $z \in S_y$ . Nodes in  $tree_y^{z,(y)}$ , utilize link  $(y, z)$ , when the service is located at node  $y$  and nodes in  $tree_y^{z,(y)}$  (the rest of the network nodes) do not. If the service moves to node  $z$ , then a subset of  $tree_y^{z,(y)}$  starts utilizing link  $(y, z)$ , while the remaining nodes (of  $tree_y^{z,(y)}$ ) do not utilize link  $(y, z)$ . Let the latter set of nodes be denoted as  $\Phi_{y \rightarrow z}$ . Therefore,  $tree_y^{z,(y)} = tree_y^{z,(z)} \cup \Phi_{y \rightarrow z}$ , while, for any node  $u \in \Phi_{y \rightarrow z}$ ,  $d_{u,z}^{(z)} < d_{u,y}^{(y)} + d_{y,z}$  is satisfied. A more formal definition of  $\Phi_{y \rightarrow z}$  is the following,

$$\Phi_{y \rightarrow z} = \{u: u \in tree_y^{z,(y)} \text{ and } d_{u,z}^{(z)} < d_{u,y}^{(y)} + d_{y,z}\}. \quad (4)$$

$\Phi_{y \rightarrow z}$  does not include those nodes  $u \in tree_y^{z,(y)}$ , for which  $d_{u,z}^{(z)} = d_{u,y}^{(y)} + d_{y,z}$ . It is assumed here that these nodes continue to utilize the same (shortest) path that they used to utilize (when the service was located at node  $y$ ), with the addition of link  $(y, z)$  (when the service moves to node  $z$ ).

Based on the aforementioned definition of  $\Phi_{y \rightarrow z}$ , it is easy to derive that,

$$tree_y^{z,(y)} = tree_y^{z,(z)} \cup \Phi_{y \rightarrow z}, \quad (5)$$

$$tree_z^{y,(z)} = tree_z^{y,(y)} \cup \Phi_{y \rightarrow z}. \quad (6)$$

**Lemma 1**  $C_z^{(z)} < C_y^{(y)}$  is satisfied, iff,

$$d_{y,z} \left( \Lambda_y^{y,(y)} - \Lambda_y^{z,(z)} \right) > \sum_{u \in \Phi_{y \rightarrow z}} \lambda_u (d_{u,z}^{(z)} - d_{u,y}^{(y)}). \quad (7)$$

**Proof:** According to equations (1), (5) and (6), and since,  $y \in S_z$ ,  $d_{y,z}^{(z)} = d_{y,z}^{(y)} = d_{y,z}$ , then,  $C_z^{(z)} - C_y^{(y)} = d_{y,z} \left( \Lambda_y^{z,(z)} - \Lambda_y^{y,(y)} \right) + \sum_{u \in \Phi_{y \rightarrow z}} \lambda_u (d_{u,z}^{(z)} - d_{u,y}^{(y)})$ . In order for  $C_z^{(z)} < C_y^{(y)}$  to be satisfied, it suffices that  $d_{y,z} \left( \Lambda_y^{z,(z)} - \Lambda_y^{y,(y)} \right) + \sum_{u \in \Phi_{y \rightarrow z}} \lambda_u (d_{u,z}^{(z)} - d_{u,y}^{(y)}) < 0$ , and the lemma is proved. ■

In view of Lemma 1 the following service migration criterion is established.

*Criterion B:* Service movement  $y \rightarrow z$  takes place, if  $d_{y,z} \left( \Lambda_y^{z,(y)} - \Lambda_y^{z,(z)} \right) > \sum_{u \in \Phi_{y \rightarrow z}} \lambda_u \left( d_{u,z}^{(z)} - d_{u,y}^{(y)} \right)$  is satisfied.

The definition of Criterion B and the cost reducing triggering of the movement under it allows for the following lemma.

**Lemma 2** *A service movement under Criterion B terminates at the end of a monotonically cost decreasing path.*

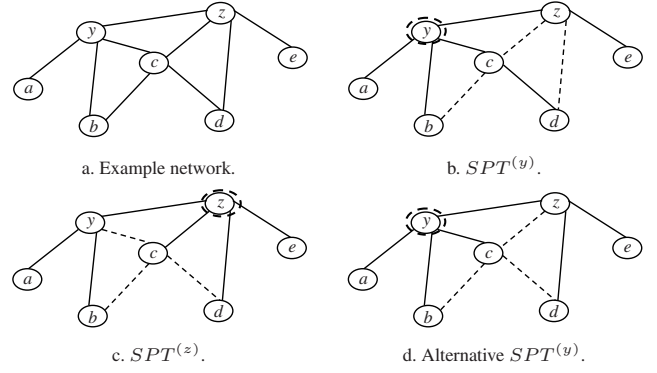
In view of Lemma 2, it is clear that Criterion B is an improvement over Criterion A that is not able to move the service until the end of a cost decreasing path (unless the unique SPT property is satisfied). This improvement is possible at the cost of requiring some non-local topology information (e.g.,  $\Phi_{y \rightarrow z}$ ,  $\lambda_u$ ,  $d_{u,z}^{(z)} - d_{u,y}^{(y)}$ ), which was not needed for the application of Criterion A. The consideration of general topologies of equal link weights in the following section allows for further elaboration on Criterion B and the subsequent introduction of Criterion  $B_e$  that is based on local information.

## 4 Migration in Equal Link Weight Topologies

In this section the focus is on general topologies with equal link weights (for simplicity, the weight of each link is considered equal to 1). Normally these topologies allow for multiple SPTs depending on the case (e.g., tree topologies is an exception due to the existence of a unique SPT regardless of the link weights). Depending on the topology, it is also possible that for a certain root  $x$ , more than one  $SPT^{(x)}$  can be drawn. Consider, for example, the network depicted in Figure 2.a:  $SPT^{(y)}$  may be either the one depicted in Figure 2.b, or the one depicted in Figure 2.d.

The increased number of SPTs in a network allows for the existence of numerous alternative shortest paths in the event of a certain service movement. This is the main limitation of Criterion A, as already discussed in Section 2, and therefore, equal link weight topologies may be considered as a worst case scenario for Criterion A. As it will be shown next, a new form of Criterion B can be derived that is based on local information (i.e., knowledge of  $\Phi_{y \rightarrow z}$ ,  $\lambda_u$  and  $d_{u,z}^{(z)} - d_{u,y}^{(y)}$ , is no longer required) provided that a certain *2-phase polling mechanism* is employed.

The proposed polling mechanism consists of two phases. During the *first polling phase* the service (tentatively) moves to node  $z$  from node  $y$ . This service movement is denoted as  $(y \rightarrow z)^1$  to emphasize the fact that it corresponds to the first polling phase. When the service is located



**Figure 2. Shortest path trees for the example network presented in Figure 2.a, for the first and the second polling phase.**

at node  $z$ ,  $SPT^{(z)}$  is formed and therefore, knowledge of  $\Lambda_y^{z,(z)}$  (denoted now as  $\left( \Lambda_y^{z,(z)} \right)^1$  since it corresponds to the first polling phase) is now available using a suitable monitoring mechanism, [10]. Afterwards, the service moves back to node  $y$  (i.e., service movement  $(z \rightarrow y)^1$ ) and this second step concludes the first polling phase. The second polling phase is similar to the first one apart from some proper adaptation of the notations (e.g., the service movement  $y \rightarrow z$  is now denoted as  $(y \rightarrow z)^2$ ).

### 4.1 Criterion $B_e$

The main aim in the rest of this subsection is to simplify the condition of Criterion B by proving that for equal link weight topologies, at the end of the 2-step polling mechanism, the right term of the criterion's inequality reduces to 0. This is possible by considering set  $\Theta_{y \rightarrow z}^{-1}$  and set  $\Theta_{y \rightarrow z}^0$ , such that  $\Phi_{y \rightarrow z} = \Theta_{y \rightarrow z}^{-1} \cup \Theta_{y \rightarrow z}^0$ , and  $\Theta_{y \rightarrow z}^{-1} \cap \Theta_{y \rightarrow z}^0 = \emptyset$ , where, (a) if a node  $u \in \Theta_{y \rightarrow z}^{-1}$ , then  $d_{u,z}^{(z)} - d_{u,y}^{(y)} = -1$ ; and (b) if a node  $u \in \Theta_{y \rightarrow z}^0$ , then  $d_{u,z}^{(z)} - d_{u,y}^{(y)} = 0$ . Note that the difference  $d_{u,z}^{(z)} - d_{u,y}^{(y)}$  can take no other values than  $-1$  or  $0$ , since (a)  $u \in \Phi_{y \rightarrow z}$ ; (b) all link weights are equal to 1; and (c) any service movement takes place among neighbor nodes. An example is illustrated in Figure 2. In Figure 2.b the service is located at node  $y$  and when it moves to node  $z$  (as depicted in Figure 2.c),  $\Theta_{y \rightarrow z}^{-1} = \{d\}$  (since the distance between node  $d$  and the service node decreases, or,  $d_{d,z}^{(z)} = d_{d,y}^{(y)} - 1$ , when the service moves to node  $z$  from node  $y$ ) and  $\Theta_{y \rightarrow z}^0 = \{c\}$  (since the distance between node  $d$  and the service node remains the same, or,  $d_{d,z}^{(z)} = d_{d,y}^{(y)}$ , when the service moves to node  $z$  from node  $y$ ). Note that for node  $b$ ,  $d_{b,z}^{(z)} = d_{b,y}^{(y)} + 1$ , and therefore,  $b \notin \Phi_{y \rightarrow z}$ . As it was mentioned in Section 3, node  $b$  will



utilize the shortest path  $\{(b, y), (y, z)\}$ , instead of the shortest path  $\{(b, c), (c, z)\}$ , in order to satisfy the need for as few changes as possible in the network.

**Lemma 3** For the service movement  $(y \rightarrow z)^2$ ,  $\Theta_{(y \rightarrow z)^2}^{-1} = \emptyset$ .

**Proof:** When the service movement  $(z \rightarrow y)^1$  takes place, it can be shown that  $\Phi_{(z \rightarrow y)^1} \cap \Theta_{(y \rightarrow z)^1}^{-1} = \emptyset$ . Actually, by definition, for any  $u \in \Theta_{(y \rightarrow z)^1}^{-1}$ ,  $d_{u,z}^{(z)} - d_{u,y}^{(y)} = -1$ , and for any node  $u \in \Phi_{(z \rightarrow y)^1}$ ,  $d_{u,y}^{(y)} < d_{u,z}^{(z)} + 1$ , or,  $d_{u,z}^{(z)} - d_{u,y}^{(y)} > -1$ . Obviously, if  $u \in \Phi_{(z \rightarrow y)^1}$ , then  $u \notin \Theta_{(y \rightarrow z)^1}^{-1}$ , and vice versa. Therefore,  $\Phi_{(z \rightarrow y)^1} \cap \Theta_{(y \rightarrow z)^1}^{-1} = \emptyset$ . Since  $(tree_y^{z,(y)})^1 = (tree_y^{z,(z)})^1 \cup \Phi_{(z \rightarrow y)^1}$ , and  $(tree_y^{z,(z)})^1 \cap \Theta_{(y \rightarrow z)^1}^{-1} = \emptyset$ , it is evident that  $(tree_y^{z,(y)})^1 \cap \Theta_{(y \rightarrow z)^1}^{-1} = \emptyset$ . Obviously, there is no node  $u \in (tree_y^{z,(y)})^1$ , such that  $d_{u,z}^{(z)} - d_{u,y}^{(y)} = -1$ . Consequently, by definition,  $\Theta_{(y \rightarrow z)^2}^{-1} = \emptyset$ , and the lemma is proved. ■

Figure 2 illustrates such an example regarding the network depicted in Figure 2.a. Assuming the service initially located at node  $y$  and the corresponding  $SPT^{(y)}$  as depicted in Figure 2.b, when the service moves to node  $z$ ,  $SPT^{(z)}$  is the one depicted in Figure 2.c. For this case,  $\Phi_{(y \rightarrow z)^1} = \{c, d\}$ ,  $\Theta_{(y \rightarrow z)^1}^{-1} = \{d\}$ , and  $\Theta_{(y \rightarrow z)^1}^0 = \{c\}$ . When the service returns to node  $y$  (i.e., service movement  $(z \rightarrow y)^1$ ), a new SPT is created (i.e.,  $SPT^{(y)}$ ), as depicted in Figure 2.d. As it was expected,  $\Phi_{(z \rightarrow y)^1} = \{c\}$ , since there is no reason for node  $d \in \Theta_{(y \rightarrow z)^1}^{-1}$ , to stop utilizing the new (and of lower cost) shortest path towards the (new) service node  $z$ . When the second polling phase takes place (i.e., service movement  $(y \rightarrow z)^2$ ), it is obvious that  $\Phi_{(y \rightarrow z)^2} = \{c\}$ , and  $\Theta_{(y \rightarrow z)^2}^{-1} = \emptyset$ .

The following migration criterion can be used in general topologies with equal link weights.

**Criterion  $B_e$ :** In a network of equal link weights, a service movement  $y \rightarrow z$  takes place, if  $(\Lambda_y^{y,(y)})^2 > (\Lambda_y^{z,(z)})^2$  is satisfied.

## 4.2 The Hybrid Scheme

Criterion  $B_e$  can be applied in general and of equal link weight topologies without any requirement for non-local information that is an important property for the considered network environments. On the other hand, the employment of the 2-phase polling mechanism introduces a certain overhead. It is possible to reduce this overhead by considering a *hybrid* combination scheme of both Criterion  $B_e$  and

**Table 1. Results.**

	RG		Grid	
	Time	$\alpha$	Time	$\alpha$
$B_e$	8	1.087	104	1
$A/B_e$	7	1	26	1

Criterion  $A$ . Under this scheme the service migrates under Criterion  $A$ . When it stops moving under Criterion  $A$ , Criterion  $B_e$  is used to decide if a next service movement is possible. If a service movement is decided under Criterion  $B_e$ , the service moves to the particular node and Criterion  $A$  is resumed for the subsequent service movements as before. Service migration terminates by the time that no service movement is possible under Criterion  $B_e$ . This approach allows for the reduction of the overall cost, when compared to the case that only Criterion  $A$  is used, and for fewer service movements, when compared to the case that only Criterion  $B_e$  is used.

In order to demonstrate the benefit when using the aforementioned hybrid scheme, simulation results (of a single run) for both cases (Criterion  $B_e$  and the hybrid scheme) are presented (these results are preliminary ones corresponding to ongoing work and will be further extended in future extensions). Two topologies of 100 nodes are considered and the aggregate service demands for all network nodes are assumed given (no need for a monitoring mechanism). The first topology considered is the outcome of the *random graph* model (RG), while the second is a grid topology of  $4 \times 25$  dimension. The diameter of the first topology is equal to 8 and of the second is equal to 27 (the diameter corresponds to an upper bound of the number of hops required for a service to move until the optimal position). Clearly, the objective is for the service (artificially located as far away as possible from the optimal position in the initial setup) to move to the optimal position. Let  $C_{opt}^{(opt)}$  denote the cost when the service is located at the optimal position. Consequently, the service moves to the optimal position (e.g., node  $u$ ), when  $\alpha = 1$ , for  $\alpha = C_u^{(u)} / C_{opt}^{(opt)}$ .

In Table 1, it is obvious that under Criterion  $B_e$ , the service is capable of moving close to the optimal position ( $\alpha = 1$  or  $\alpha \approx 1$  for all cases) in 8 time units in the random graph case and 104 time units in the grid case. When the hybrid scheme is applied, it is evident that the overall time is reduced especially for the case of grids. This can be explained based on the different size of the diameter for both networks. Actually, when the polling mechanism is used, this has a significant effect on the grid topology (due to the significantly larger diameter when compared to the random graph). On the other hand, when the hybrid scheme is applied, the polling mechanism may be avoided for some movements (due to the employment of Criterion  $A$ ).

An upper bound regarding time units until the service movements to the optimal (or to a close to the optimal) position is  $D \times \bar{k}$ , where  $D$  is the network diameter and  $\bar{k}$  the average number of network nodes. This upper bound assumes that Criterion  $B_e$  is applied in every step. The best case under the hybrid scheme (i.e., only Criterion  $A$  is applied and therefore, the polling mechanism is not used), allows for the reduction of the previous bound to  $D$ .

## 5 Summary and Conclusions

In this paper, the efficient service placement problem for general topologies is studied. This problem, that is identical to the 1-median problem, is hard to be solved since existing approaches require global information and increased number of messages. As a result these approaches are not suitable for large-scale and/or dynamic environments. In order to cope with this inefficiency, the idea of service migration, [10], is exploited in this paper and a previously proposed service migration Criterion  $A$ , [10], was extended here to cope with certain inefficiencies.

This motivated the introduction of service migration Criterion  $B$  proposed here, aiming at general topologies. Equal link weight topologies were also considered here, corresponding to a worst case scenario for Criterion  $A$  due to the increased number of alternative shortest paths in the network, allowing, at the same time, for the derivation of a simplified form of Criterion  $B$ , i.e. Criterion  $B_e$ . Criterion  $B_e$  requires local information (as opposed to Criterion  $B$  that turned on to require non-local network information) to decide on the service movement and as it was analytically shown, it is capable of moving the service along and till the end of a cost decreasing path at the cost of a two phase polling mechanism. In the sequel, a hybrid scheme, combining both Criterion  $A$  and Criterion  $B_e$ , was proposed resulting in reduction of the overhead introduced by the polling mechanism, the claims being supported by simulation results.

In conclusion, service migration is shown here that is not limited in tree topologies, [10], but can be also applied to general cases. The proposed new criteria in this paper introduce some overhead (due to the polling mechanism) but on the other hand they rely on local information and are fully distributed which is a plus with respect to scalability in large network environments, [13]. Further future work will consider criteria for migration of multiple service facilities in the network and will provide for further simulation results.

## 6. Acknowledgments

This work has been supported in part by the project Autonomic Network Architecture (ANA) (IST-27489), which

is funded by the IST FET Program of the European Commission, and in part by the NoE CONTENT (IST-384239).

## References

- [1] I. Baev and R. Rajaraman. Approximation algorithms for data placement in arbitrary networks. In *Proceedings of the 12th Annual Symposium on Discrete Algorithms (ACM-SIAM SODA)*, 2001, pages 661–670.
- [2] D. Bertsekas and R. Gallager. *Data networks*. 2nd edition, Prentice-Hall, Inc., 1992.
- [3] O. Kariv and S. Hakimi. An algorithmic approach to network location problems, ii: The p-medians. *SIAM Journal on Applied Mathematics*, 37, 3 (1979), pages 539–560.
- [4] J.-H. Lin and J. Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 1992, 44, pages 245–249.
- [5] C. G. P. M. R. Korupolu and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998, pages 1–10.
- [6] C. G. P. M. R. Korupolu and R. Rajaraman. Placement algorithms for hierarchical cooperative caching. In *Proceedings of the 10th Annual Symposium on Discrete Algorithms (ACM-SIAM SODA)*, 1999, pages 586–595.
- [7] P. Mirchandani and R. Francis. *Discrete Location Theory*. Cambridge University Press, 1990.
- [8] K. O. I. S. A. B. N. Laoutaris, G. Smaragdakis. Distributed placement of service facilities in large-scale networks. In *IEEE INFOCOM'07, May 6-12, 2007, Anchorage, Alaska*, 2007.
- [9] V. Z. N. Laoutaris and I. Stavrakakis. Joint object placement and node dimensioning for internet content distribution. *Information Processing Letters*, Vol. 89, No. 6, 2004, pages 273–279.
- [10] K. Oikonomou and I. Stavrakakis. Scalable service migration: The tree topology case. In *The Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2006)*, Lipari, Italy, June 14-17, 2006.
- [11] A. Tamir. An o(pn) algorithm for p-median and related problems on tree graphs. *Operations Research Letters*, 1996, 19, pages 59–64.
- [12] R. K. K. M. A. M. V. Arya, N. Garg and V. Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings of the 33rd Annual Symposium on Theory of Computing (ACM STOC)*, 2001, pages 11–29.
- [13] G. Wittenburg and J. Schiller. A Survey of Current Directions in Service Placement in Mobile Ad-hoc Networks. In *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '08)*, Hong Kong, Mar. 2008.
- [14] L. Yamamoto and G. Leduc. Autonomous reflectors over active networks: towards seamless group communication. *AISB journal, special issue on agent technology*, 2001, pages 125–146.