Scalable Service Migration in Autonomic Network Environments

Konstantinos Oikonomou, *Member, IEEE,* and Ioannis Stavrakakis, *Fellow, IEEE*

***Abstract*— Service placement is a key problem in communication networks as it determines how efficiently the user service demands are supported. This problem has been traditionally approached through the formulation and resolution of large optimization problems requiring *global knowledge* and a continuous recalculation of the solution in case of network changes. Such approaches are not suitable for large-scale and dynamic network environments. In this paper, the problem of determining the optimal location of a service facility is revisited and addressed in a way that is both *scalable* and deals inherently with network dynamicity. In particular, service migration which enables service facilities to move between neighbor nodes towards more communication cost-effective positions, is based on *local information*. The *migration policies* proposed in this work are analytically shown to be capable of moving a service facility between neighbor nodes in a way that the cost of service provision is reduced and – under certain conditions – the service facility reaches the optimal (cost minimizing) location, and locks in there as long as the environment does not change; as network conditions change, the migration process is automatically resumed, thus, naturally responding to network dynamicity under certain conditions. The analytical findings of this work are also supported by simulation results that shed some additional light on the behavior and effectiveness of the proposed policies.**

***Index Terms*— Service Placement, Service Migration, Autonomic Networks, Scalability.**

I. Introduction

INTERNET globalization and expansion make the service placement problem a challenging one and necessitate a careful selection of the location of the service facilities (a *facility* being a service provisioning infrastructure), aiming at bringing the service provision points close to the demand in order to minimize communication *costs* (i.e., resource consumption) and enhance the Quality of Service (QoS) of the provided service. Due to the recent technological changes (e.g., powerful machines and services have proliferated), 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 transformed into a problem of placing the one or more service facilities in one of the numerous network nodes that are now capable of hosting services. Peer-to-peer networks, cloud computing, content distributions networks, software updates and patches and sensor networks are examples of such modern environments.

The problem of determining the optimal service placement has been studied in the past in areas such as transportation and supply networks [1], and has been approached through the formulation and solution of large optimization problems (NP-hard) requiring *global knowledge*, as for instance is the case with the k-median problem [2]. Such approaches requiring global knowledge and a continuous recalculation of the solution in case of network changes; *do not scale* and are not suitable for dynamic network environments, such as those considered in this work. Instead, approaches based on *local information* should be adopted, despite the fact that they might not be able to guarantee optimality all the time (near-optimal solutions).

In this paper, the problem of determining the optimal location of a service facility is revisited and addressed in a way that is both scalable and deals inherently with network dynamicity. The approach advocated in this paper – parts of it initially presented in [3] and [4] – is that of moving a service facility among neighbor nodes by utilizing local information, in a such way that the cost of service provision is reduced and the service facility reaches under certain conditions the optimal (cost minimizing) location, and locks in there as long as the environment does not change; as network conditions change, the migration process is automatically resumed, thus, naturally responding to network dynamicity.

The first proposed policy (referred to as *Migration Policy* S) is analytically shown to be capable of moving the facilities along a monotonically cost decreasing path in the network. For special cases of topologies such as trees, it is analytically shown that under Migration Policy S a single facility moves until it reaches the optimal position (i.e., the node at which the overall cost is minimized). In the general case, service facility migration under Migration Policy S, allows for overall cost reduction, but it may fail to move the facilities until the end of a monotonically cost decreasing path, mostly due to unforeseen *shortcuts* (i.e., alternative shortest paths utilized by some nodes to reach a certain facility after a facility movement). The potential cost reduction that is due to the aforementioned shortcuts is not taken into account under Migration Policy S, thus certain facility movements that would allow for further cost reduction are not detected and thus not implemented.

The aforementioned limitation is overcome in the case of a single facility and for topologies of equal link weights, while still utilizing only local information. Note that topologies with equal link weights and a single service facility are not uncommon. For such environments, a new policy (referred to

This work has been supported in part by the IST-FIRE project Autonomic Network Architecture (ANA) (IST-27489) and the IST-FET project SOCIALNETS (IST-217141), funded by the European Commission.

Konstantinos Oikonomou is with the Department of Informatics, Ionian University, Corfu, Greece. Address: Tsirigori Square 7, 49100 Corfu, Greece. Phone: +30 26610 87708, Fax: +30 26610 48491, E-mail: okon@ionio.gr.

Ioannis Stavrakakis is with the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens, Greece. Address: Panepistimiopolis, Ilissia, 157 84, Athens, Greece. Phone: +30 210 7275315, Fax: +30 210 7275333, E-mail: ioannis@di.uoa.gr.

as *Migration Policy* E) is proposed that moves the facility until the end of a monotonically cost decreasing path, provided that tentative movements to the one-hop neighbor nodes are allowed.

The additional overhead due to the tentative facility movements under Migration Policy E (when compared to Migration Policy S), motivates the introduction of a hybrid policy (referred to as *Migration Policy* H) that efficiently combines the Migration Policies S and E, in the case of equal link topologies and a single facility. Under Migration Policy H, a single facility moves until no further movement is possible under Migration Policy S. Then, Migration Policy E takes over and moves the facility until no further movement is possible. The first part of facility movement benefits from the guaranteed cost reduction under Migration Policy S and the almost negligible overhead. The second part moves the facility towards positions of even smaller overall cost (i.e., the end of the monotonically cost decreasing path), at the expense of the extra overhead due to the tentative facility movements.

The analytical findings of this work are also supported by simulation results. In addition, simulation results provide for further insight on the behavior of the proposed policies and, particularly, illustrate their effectiveness for cases not captured by the analysis. More specifically, it is shown that in realistic topologies, the overhead (due to tentative facility movements) under Migration Policies E and H can be prohibitively large and therefore, Migration Policy S becomes the obvious alternative. This is also the case in dynamic environments due to the latter's capability to move – almost immediately – service facilities to more effective positions.

Section II presents related work and in Section III the service placement problem is described in detail. Migration Policy S is presented and studied in Section IV while Migration Policies E and H are presented and studied in Section V. Section VI presents simulation results and the conclusions are drawn in Section VII.

II. Related Work

The service placement problem has been addressed in the past particularly in the area of transportation, supply networks etc., most of these works surveyed in [1]. It has been shown that the optimal solution of placing k service facilities in a network (i.e., the k-median problem) is an NP-hard problem for general graphs [2]. Even in the case of undirected tree topologies, the complexity remains as high as $O(kN)$ [5] (N is the number of nodes in the network). In order to deal with the increased complexity, several near-optimal approaches have been proposed that can be categorized as either centralized or distributed [6]. The centralized approaches focus on greedy heuristics [7], [8], [9], [10], on linear programming [11], [12], [13], on primal-dual [14], [15], local search [16], [17], and other techniques [18], [19], [20], [21], [22], [23] that have been proposed in the past to deal with the increased complexity of the service placement problem. However, all of them require global knowledge of the network topology and demands, and this information is generally not available in the large-scale network environment considered in this paper.

The focus in this work is on distributed approaches being based on local information (as opposed to the aforementioned centralized local-search-based approaches attempting to assume local minima of performance) in order to avoid scalability problems introduced by global knowledge requirements. To the best of the authors' knowledge, the first distributed facility location was described by Jain and Vazirani [14]. Moscibroda and Wattenhofer [24] proposed the first distributed approach. Their work focuses on primal-dual techniques – in order to derive worst-case performance bounds – which are difficult to implement (e.g., impractical communication model) compared to the work presented in this paper. A recent work by Krivitski et al. [25] proposes a distributed hill-climbing algorithm based on local majority votes and used by nodes to agree on the next step of the algorithm. The overall overhead is kept low by avoiding unnecessary votes. The algorithm converges to the optimal solution, as shown using simulation results. This approach is different from the one proposed here (evaluated both through analysis and simulations) since no majority voting is considered and the need for local information is almost negligible.

The closest works to the one presented here are [26] and [27]. Ragusa et al. [26] propose a heuristic approach for partitioning the network into a number of k clusters self-managed through mobile agents. Each agent migrates to more efficient positions within the cluster after deriving the median position in a centralized manner. Clusters may be further partitioned or merged in response to dynamic network changes based on empirically selected threshold values. The second approach, proposed by Laoutaris et al. [27] solves the service placement problem in a distributed manner reusing existing near-optimal centralized approaches inside suitably defined r-balls (i.e., network areas of r hops away from each service facility). As service facilities move inside the network, r-balls are partitioned or merged based on the outcome of the centralized approaches. The main difference between [26], [27] and the work presented here, amounts to the requirement for local information in order to select more efficient positions for the mobile agents in [26] and to incorporate classical centralized solution approaches in [27]. Instead, the proposed service migration approach relies on strictly local information; this local information is passively collected by the service facility itself, as opposed to requiring the deployment of a mechanism that provides both the demand and topology information within each cluster as in [26] and [27].

Several other approaches have also been proposed, specific to the considered applications, including intermediate caching [28], reflectors' deployment [29], online version [30], placement based on mobility [31], content distribution [32], replication in overlays [33], service discovery in mobile ad hoc networks [34], gateway placement [35], sensor networks [36], [37], replica placement updates [39], distributed shared platform [38].

III. Problem Statement

The network topology is represented by a connected undirected graph $G(V, E)$, where V is the set of nodes and E

the set of links among them. Let $|X|$ denote the size of a particular set X. Let N be the number of nodes in the network or $N = |V|$. Let S_v denote the set of neighbor nodes of node v (i.e., nodes having a link with node v). Let (u, v) denote the link between two neighbor nodes u and v; each such link is assigned a positive value referred to as *weight* and denoted as $w(u, v)$. For a given graph $G(V, E)$, let $d(x, y)$ denote the *distance* between node x and node y, corresponding to the summation of the weights of the links along a *shortest path* between these nodes (for the same node v, $d(x, x) = w(x, x) = 0$); alternatively, $d(x, y)$ is referred to as the *traveling cost* between nodes x and y. In this paper, a facility is considered to be hosted by one of the network nodes. It is also possible that the service provisioning is replicated – to cope with a higher demand or for increased reliability – and, thus, more that one facility may be deployed. Without any loss of generality, only one service – and the associated one or multiple service facilities – is considered in this paper.

Let the mean rate at which data packets associated with a particular node v are transferred through the network be denoted by λ_v (i.e., service demands). Let K_t denote the set of facility nodes (i.e., nodes hosting a facility corresponding to a certain service) at time t. In the sequel, t will take discrete values corresponding to facility movements.

Between a given facility node and any other node in the network, a shortest path route is assumed to be established by the underlying employed routing protocol, [40]. Eventually, a shortest path tree is created, rooted at the particular facility node and including all network nodes. Such a tree (depicted with dense lines) rooted at the facility node 0 (marked by a dotted hexagonal) is shown in Figure 1.a, where the service demands and link weights are set to 1 to facilitate the discussion. In general, a shortest path tree associated with a given root node is not unique. When the number of facilities is greater than one, then a forest of shortest path trees is created, with each tree rooted at the corresponding facility node (e.g., as depicted in Figure 1.b). Let T_t denote the set of all possible shortest path trees in a network at time t. Let $T_t(x)$ denote the subset of T_t (i.e., $T_t(x) \subseteq T_t$) containing the shortest path trees rooted at node x. Let T_t^x be the shortest path tree in $T_t(x)$ over which data corresponding to the nodes' service demands are forwarded towards the facility node x. $v \in T_t^x$ will indicate that node v is served by facility node x.

Let $x \xrightarrow{t} y$ denote the *facility movement* from node x to its neighbor node y initiated at time t; at time t $(t + 1)$ the facility is located at node x (y). Assume that $K_0 = \{0\}$ and T_0^0 is the shortest path tree depicted in Figure 1.a. Assume that $K_1 = \{1\}$ and T_1^1 is the shortest path tree depicted in Figure 1.c after a facility movement $0 \xrightarrow{0} 1$. It is reasonable to assume that any routing protocol would try to minimize the overhead introduced by such a facility movement by preserving previously established shortest paths and not switching to new ones, provided that the previous ones are not worse than any new one. This *migration rule* will be adopted here and ensures that the new shortest path tree T_1^1 would be the one depicted in Figure 1.c and not the one depicted in Figure 1.d (both belong to $T_1(1)$) or any other. To illustrate this rule further, consider node 7 that forwards data towards the facility node 0 over path $\{(7, 8), (8, 0)\}$, as depicted in Figure 1.a. After the service moves to node 1, node 7 would – based on the above rule – utilize path $\{(7, 8), (8, 0), (0, 1)\}$, instead of the equal cost path $\{(7, 6), (6, 0), (0, 1)\}$.

Let $C_t(x)$ denote the cost incurred by facility x for serving nodes v at time t, for all $v \in T_t^x$. Clearly,

$$C_t(x) = \sum_{\forall v \in T_t^x} \lambda_v d(v, x). \tag{1}$$

The *overall cost* over all facilities in the network at time t, denoted as C_t, is given by, $C_t = \sum_{\forall x \in K_t} C_t(x) = \sum_{\forall x \in K_t} \sum_{\forall v \in T_t^x} \lambda_v d(v, x)$. Assuming fixed network topology and service demands, it is evident that the (optimal) set of facility locations for which cost minimization is achieved (denoted by K) does not depend on time t, and the same holds true for the corresponding *minimum cost* defined as C. Let $a_t = \frac{C_t}{C}$ be defined as the *approximation ratio* of the cost induced at time t when the set of facility nodes is K_t, over the minimum (optimal) one; the closer the value of a_t to 1, the closer the induced cost at time t to the optimal one.

The optimal set K and the resulting minimal cost C can be determined by solving the previously mentioned k-median problem, $(k = |K|)$. This difficult and large optimization problem cannot be afforded in the large-scale and dynamic network environments considered here, where the network topology is subject to frequent changes requiring the recalculation of the (expensive) k-median solution. In the sequel, services are migrated in order to exploit information locally available at the facility nodes.

IV. Service Migration Based on Strictly Local Information

Strictly local information refers to information that is available only at a particular node. Based on such information, a migration policy is proposed in this section to reduce the cost after moving the facilities to neighbor nodes.

Assume a service facility is located at node x at time t. There exist a number of neighbor nodes S_x over which data associated with service demands of all nodes $v \in T_t^x \setminus \{x\}$, are forwarded to the particular facility. In the example depicted in Figure 1.a, data associated with the service demands towards the facility node 0, are forwarded over link $(0, 1)$ for nodes 1 and 2, over link $(0, 6)$ for nodes 3, 4, 5 and 6, and over link $(0, 8)$ for nodes 7, 8 and 9. That is, nodes of a certain subtree of T_t^x forward data associated with their own service demands through some node y, $y \in S_x$. Let $I^y(T_t^x)$ denote the particular subtree, which is also a tree of root node y. In the previous example, there are three such subtrees denoted by $I^1(T_t^0)$, $I^6(T_t^0)$ and $I^8(T_t^0)$, as shown in Figure 1.a.

Let $\Lambda\left(I^y(T_t^x)\right)$ denote the *aggregate service demands* that are forwarded to the facility node x through link (x, y) (for some neighbor node $y \in S_x$, and $y \in T_t^x$) over subtree $I^y(T_t^x)$. $\Lambda\left(I^y(T_t^x)\right)$ is equal to the summation of the service demands of the individual nodes of the corresponding subtree, or:

$$\Lambda\left(I^y(T_t^x)\right) = \sum_{\forall v \in I^y(T_t^x)} \lambda_v. \tag{2}$$

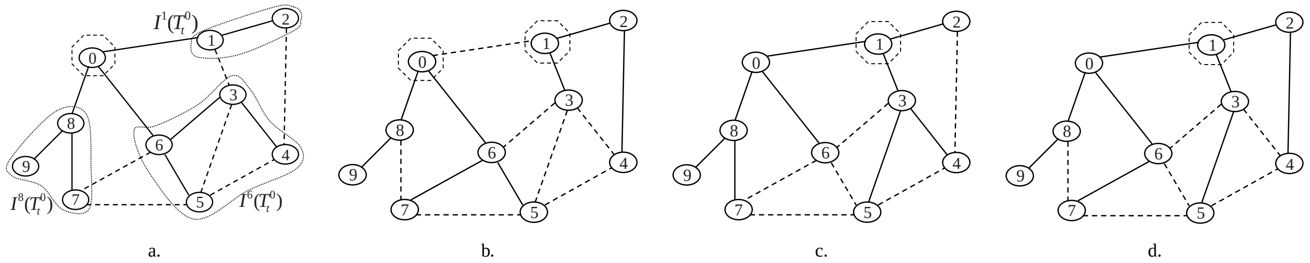


Fig. 1. Shortest path trees and subtrees for an example network.

$\Lambda\left(I^y(T_t^x)\right)$ can be available to facility node x using a monitoring mechanism that captures the incoming and outgoing packets or, in case λ_v is known to node v, by communicating these values to x (e.g., through piggybacking). It will be assumed that each facility node x has knowledge of $\Lambda(I^y(T_t^x))$ for all neighbor nodes $y \in S_x$. This locally available information will be utilized by the proposed service migration policy.

A. Service Migration for a Single Facility

Consider the case of one facility located at node x at time t. The key question for service migration is to establish whether a cost reduction is achieved by moving the facility to node y. Since – according to Equation (1) – the previous requires global information, the main challenge is to derive a condition that would be based on information that is locally available at the facility node x, such as $\Lambda(I^y(T_t^x))$.

Let $C_{t+1}^{T_t^x}(y)$ denote a *hypothetical cost* assuming that (a) the facility moves to node y at time $t+1$; and (b) the corresponding shortest path tree over which data are forwarded towards the facility node y (which should have been T_{t+1}^y, if facility movement $x \xrightarrow{t} y$ had actually taken place) remains the current one (i.e., T_t^x). For this hypothetical cost, let the distance between any node v that is served by facility y over the shortest path tree T_t^x be denoted by $d^{T_t^x}(v,y)$ instead of $d(v,y)$. Equivalently, $C_{t+1}^{T_t^x}(y) = \sum_{\forall v \in T_t^x} \lambda_v d^{T_t^x}(v,y)$. Note that in general shortest path trees are different for different roots (i.e., $T_{t+1}^y \neq T_t^x$), except for the special case of topologies with unique shortest path trees [40]. *Unique shortest path tree topologies* are those for which $T_t(x) = T_t(y)$, for all pairs of nodes $x, y \in V$, at any time t. The following lemmas are the basis for the migration policy presented later.

Lemma 1: For a single service facility in a network located at some node x at time t and some neighbor node $y \in S_x$, $C_{t+1}^{T_t^x}(y) \geq C_{t+1}(y)$ is satisfied (the equality holds for unique shortest path tree topologies). In addition, the difference between cost $C_{t+1}^{T_t^x}(y)$ and cost $C_t(x)$ is given by:

$$C_{t+1}^{T_t^x}(y) - C_t(x) = \Big(\Lambda\big(T_t^x \setminus I^y(T_t^x)\big) - \Lambda\big(I^y(T_t^x)\big)\Big) w(x,y). \tag{3}$$

The proof of Lemma 1 can be found in Appendix A.

The right part of Equation (3) depends on the link weight $w(x,y)$, the aggregate service demands that are forwarded to node x through node y (i.e., $\Lambda\big(I^y(T_t^x)\big)$) and the rest of the aggregate service demands that arrive through the other neighbor nodes of x (i.e., set $S_x \setminus \{y\}$) plus the service demands of node x itself (i.e., $\sum_{\forall v \in S_x \setminus \{y\}} \Lambda\big(I^v(T_t^x)\big) + \lambda_x = \Lambda\big(T_t^x \setminus I^y(T_t^x)\big)$, since $\cup_{\forall v \in S_x \setminus \{y\}} I^v(T_t^x) \cup \{x\} = T_t^x \setminus I^y(T_t^x)$). As mentioned before, both $\Lambda\big(T_t^x \setminus I^y(T_t^x)\big)$ and $\Lambda\big(I^y(T_t^x)\big)$ are locally available at node x (i.e., strictly local information).

In view of Lemma 1, two interesting observations can be made regarding the difference when the facility is located at neighbor nodes. First, the difference does not depend on the weights of the links of the network, apart from the weight of the link among them, i.e., $w(x,y)$. Second, it depends on the difference between the aggregate service demands. Consequently, global knowledge of the network (i.e., knowledge of the weights of each link and the service demands of each node in the network) is not necessary to determine differences in costs associated with neighboring facility nodes and, eventually, determine the facility node that induces the lowest cost among all neighboring nodes. Even knowledge of $w(x,y)$ is not necessary, as it is shown later in Theorem 1. What is actually required is information regarding the aggregate service demands, which can be available at the facility node.

Theorem 1: For a single service facility in a network located at some node x at time t and some neighbor node $y \in S_x$, if the facility moves to node y, then cost reduction is achieved, i.e., $C_{t+1} < C_t$, provided that $\Lambda\big(T_t^x \setminus I^y(T_t^x)\big) < \Lambda\big(I^y(T_t^x)\big)$.

The proof of Theorem 1 can be found in Appendix B.

Motivated by Theorem 1, the following migration policy is proposed, referred to hereafter as *Migration Policy S* to emphasize the use of strictly local information (i.e., S) and to distinguish it from other migration policies proposed later in this paper.

Definition of Migration Policy S: For a facility located at some node x at time t, the facility is moved from node x to some neighbor node y at time t iff $\Lambda\big(T_t^x \setminus I^y(T_t^x)\big) < \Lambda\big(I^y(T_t^x)\big)$.

According to Migration Policy S and in view of Theorem 1, it is easy to conclude that every movement of the facility results in cost reduction.

B. Multiple Facilities

The following theorem shows that under Migration Policy S, overall cost reduction is always achieved (i.e., $C_{t+1} < C_t$), whatever the number of facilities in the network.

Theorem 2: In a network of more than one facilities, if a facility located at some node x at time t moves under

Migration Policy S to some neighbor node y, then $C_{t+1} < C_t$. The proof of Theorem 2 can be found in Appendix C.

Moving a facility under Migration Policy S and achieving overall cost reduction does not necessarily mean that the facility will eventually reach the optimal position (i.e., the solution of the k-median problem) that minimizes the overall cost. This is guaranteed (as shown next) for unique shortest path tree topologies (e.g., trees) and for a single facility.

Theorem 3: In a network consisting of a unique shortest path tree, a single service facility always arrives at the optimal location under Migration Policy S, assuming a static environment.
The proof of Theorem 3 can be found in Appendix D.

Unique shortest path tree topologies (e.g., trees) are not uncommon; in fact, trees are formed frequently as a result of routing protocols in dynamic environments (e.g., mobile ad hoc networks [6]). In addition, the presence of a single service facility within certain network boundaries is also frequent. Consequently, the results of Theorem 3 apply to many real cases. Migration Policy S is also useful for environments that do not comply with the previous conditions, since it allows for cost reduction (even though not necessarily for cost minimization) based on strictly local information.

V. Service Migration Based on One-hop Local Information

Suppose that at time t a single facility is located at node x and moves to neighbor node y. If the shortest path tree of root node y is different from that of root node x (i.e., $T_{t+1}^y \neq T_t^x$), then this indicates that some nodes have preferred a *shortcut*, i.e., a shortest path towards the new facility node y that is shorter than that towards node x plus the weight $w(x,y)$ (further details about shortcuts are given next). Migration Policy S fails to capture the potential cost reduction caused by the aforementioned shortcuts and decide on a facility movement that would eventually allow for further cost reduction. The aim of this section is to overcome this limitation of Migration Policy S using information that is available at the neighbor nodes (one-hop) of a facility node by permitting tentative movements of the facility to these nodes and reducing that way the impact of the (unknown) shortcuts.

Consider the facility movement $0 \xrightarrow{t} 1$ depicted in Figures 1.a and 1.c. It easy to see that there exist nodes whose distance from the facility remains the same, or increases, or decreases. In that example, nodes 3 and 5 have chosen a shortcut to implement the new shortest path towards the (new) facility node (node 3 is a neighbor of the new facility node and node 5 utilizes the path through node 3, instead of the one through node 6). Note that when the facility was located at node 0, the existence of these shortcuts was not known to node 0; this is basically the reason why Migration Policy S is unable to foresee the corresponding cost reduction (it assumes that node 3 will utilize path $\{(3,6),(6,0),(0,1)\}$ instead of link $(3,1)$, and that node 5 will utilize path $\{(5,6),(6,0),(0,1)\}$ instead of path $\{(5,3),(3,1)\}$, which is a safe worst-case assumption).

In view of the above discussion, let $\Phi_{x \xrightarrow{t} y}$ denote the set of nodes of $T_t^x \setminus I^y(T_t^x)$ utilizing a shortcut towards the new facility node (i.e., node y at time $t+1$). Basically, the reason for some node v to utilize a shortcut is that the distance towards the new facility node y over this shortcut (i.e., $d(v,y)$) is smaller than that utilizing the previous path towards the (previous facility) node x (i.e., $d(v,x)$) plus the weight of link (x,y) (i.e., $w(x,y)$). More formally:

$$\Phi_{x \xrightarrow{t} y} = \{v : v \in T_t^x \backslash I^y(T_t^x) \text{ and } d(v,y) < d(v,x) + w(x,y)\}. \tag{4}$$

In the previous example, $\Phi_{0 \xrightarrow{t} 1} = \{3,5\}$.

A. Cost Reduction and Tentative Facility Movements

Lemma 2: For a single facility in a network and facility movement $x \xrightarrow{t} y$, $C_{t+1} < C_t$ is satisfied iff:

$$\begin{aligned}\Big(\Lambda\big(I^y(T_t^x)\big) \quad - \quad & \Lambda\big(I^x(T_{t+1}^y)\big)\Big) w(x,y) \\ & > \sum_{\forall v \in \Phi_{x \xrightarrow{t} y}} \lambda_v \big(d(v,y) - d(v,x)\big).\end{aligned} \tag{5}$$

The proof of Lemma 2 is similar to the proof of Lemma 1.

Based on the paradigm of Migration Policy S that was proposed after Theorem 1, another policy could be proposed based on Lemma 2, requiring knowledge of $\Lambda\big(I^y(T_t^x)\big)$, $\Lambda\big(I^x(T_{t+1}^y)\big)$, $w(x,y)$ and $\sum_{\forall v \in \Phi_{x \xrightarrow{t} y}} \lambda_v \big(d(v,y) - d(v,x)\big)$. $\Lambda\big(I^y(T_t^x)\big)$ and $w(x,y)$ are available at time t at node x. On the other hand, $\Lambda\big(I^x(T_{t+1}^y)\big)$, which is available at node y at time $t+1$, could be made available after a tentative facility movement to node y and then moving back to the previous facility node x before deciding if a (permanent) facility movement $x \xrightarrow{t+2} y$ should take place. However, $\sum_{\forall v \in \Phi_{x \xrightarrow{t} y}} \lambda_v \big(d(v,y) - d(v,x)\big)$ is not known either at node x or node y at any time. Eventually, the condition of Lemma 2 requires information that is not locally available.

B. Service Migration in Topologies of Equal Link Weights

The main objective of the analysis next is to simplify the right term of Lemma 2 using information that is available at least one hop away from the facility node (i.e., term $\sum_{\forall v \in \Phi_{x \xrightarrow{t} y}} \lambda_v \big(d(v,y) - d(v,x)\big)$). The objective is not to determine $\Phi_{x \xrightarrow{t} y}$ or $\big(d(v,y) - d(v,x)\big)$, which are based on non-local information, but rather to exploit the aforementioned tentative facility movements in such a way that the right term of Lemma 2 becomes obsolete (i.e., 0). This is possible in topologies with equal link weights (normalized to 1 here), as shown analytically next.

Topologies with equal link weights (e.g., $w(u,v) = 1$, $\forall (u,v) \in E$) may be considered as a worst case scenario with respect to shortcuts (provided that the topology is not a tree). In particular, equal link weights allow for an increased number of alternative paths in the network, which on the event of a facility movement are likely to be utilized (i.e., shortcuts), resulting in $\Phi_{x \xrightarrow{t} y} \neq \emptyset$, for some facility movement $x \xrightarrow{t} y$. In these topologies, $\Phi_{x \xrightarrow{t} y}$ can be analyzed further as follows.

Lemma 3: For a single facility in a network of link weights equal to 1 and a facility movement $x \xrightarrow{t} y$, for some node $y \in$

S_x, then for any node $v \in \Phi_{x \xrightarrow{t} y}$, either $d(v,y) - d(v,x) = 0$ or $d(v,y) - d(v,x) = -1$.
The proof of Lemma 3 can be found in Appendix E.

Let $\Phi^0_{x \xrightarrow{t} y}$ denote the subset of nodes $v \in \Phi_{x \xrightarrow{t} y}$ for which $d(v,y) - d(v,x) = 0$, and $\Phi^{-1}_{x \xrightarrow{t} y}$ denote the subset of nodes $v \in \Phi_{x \xrightarrow{t} y}$ for which $d(v,y) - d(v,x) = -1$. $\Phi_{x \xrightarrow{t} y} = \Phi^0_{x \xrightarrow{t} y} \cup \Phi^{-1}_{x \xrightarrow{t} y}$ and $\Phi^0_{x \xrightarrow{t} y} \cap \Phi^{-1}_{x \xrightarrow{t} y} = \emptyset$. In the example presented in Figure 1.a and the facility movement $0 \xrightarrow{t} 1$, as depicted in Figure 1.c, $\Phi^0_{0 \xrightarrow{t} 1} = \{5\}$ ($d(5,0) = 2$ and $d(5,1) = 2$) and $\Phi^{-1}_{0 \xrightarrow{t} 1} = \{3\}$ ($d(3,0) = 2$ and $d(3,1) = 1$).

It is possible to simplify the right term of Lemma 2 since by definition, for any $v \in \Phi^0_{x \xrightarrow{t} y}$, $d(v,y) - d(v,x) = 0$. The following lemma exploits tentative facility movements to allow for further simplification.

Lemma 4: For a single facility in a network of link weights equal to 1 located at node x at time t and tentative facility movements $x \xrightarrow{t} y$ and $y \xrightarrow{t+1} x$, for some node $y \in S_x$, if $x \xrightarrow{t+2} y$ takes place, then $\Phi^{-1}_{x \xrightarrow{t+2} y} = \emptyset$.
The proof of Lemma 4 can be found in Appendix F.

Theorem 4: For a single facility in a network of link weights equal to 1 located at node x at time t, cost reduction is achieved for facility movement $x \xrightarrow{t+2} y$, $y \in S_x$ iff $\Lambda\big(I^x(T^y_{t+1})\big) < \Lambda\big(I^y(T^x_{t+2})\big)$, after tentative facility movements $x \xrightarrow{t} y$ and $y \xrightarrow{t+1} x$.
The proof of Theorem 4 is based on Lemmas 2 and 4 and the fact that shortest path tree T^y_{t+3} is identical to T^y_{t+1}, since there is a single facility.

Motivated by Theorem 4, the following migration policy is proposed for topologies of equal link weights, referred to hereafter as *Migration Policy E*.

Definition of Migration Policy E: For a facility located at node x at time t, the facility is moved from node x to the neighbor node y at time $t+2$, if $\Lambda\big(I^x(T^y_{t+1})\big) < \Lambda\big(I^y(T^x_{t+2})\big)$, after tentative facility movements $x \xrightarrow{t} y$ and $y \xrightarrow{t+1} x$.

Migration Policy E is capable of moving a single facility repeatedly until the end of a monotonically cost decreasing path for topologies of equal link weights, as ensured by Theorem 4 (cost reduction is achieved iff $\Lambda\big(I^x(T^y_{t+1})\big) < \Lambda\big(I^y(T^x_{t+2})\big)$ is satisfied, or, according to the definition of Migration Policy E, iff a facility movement takes place under Migration Policy E). The previous inequality is based on aggregate service demands information that is locally available at the candidate new facility node y after a tentative facility movement to node y from the current facility node x (i.e., $\Lambda\big(I^x(T^y_{t+1})\big)$) and locally available at node x after the tentative facility movement back to node x (i.e., $\Lambda\big(I^y(T^x_{t+2})\big)$).

C. Hybrid Policy

As the tentative movements associated with Migration Policy E introduce overhead (two facility movements per neighbor node), a hybrid policy is proposed here that combines Migration Policy S and E so that tentative movements are avoided whenever possible.

Lemma 5: In topologies of a unique shortest path tree and a single facility, Migration Policies S and E permit the same facility movements. For any other topology of equal link weights, if a facility movement is permitted under Migration Policy S, then it is also permitted under Migration Policy E. The reverse is not always true.
The proof of Lemma 5 can be found in Appendix G.

This lemma motivates the following proposed (hybrid) Migration Policy H.

Definition of Migration Policy H: A facility moves in the network under Migration Policy S as far as possible, then it is moved under Migration Policy E. The facility stops if it cannot move under any of the aforementioned policies.

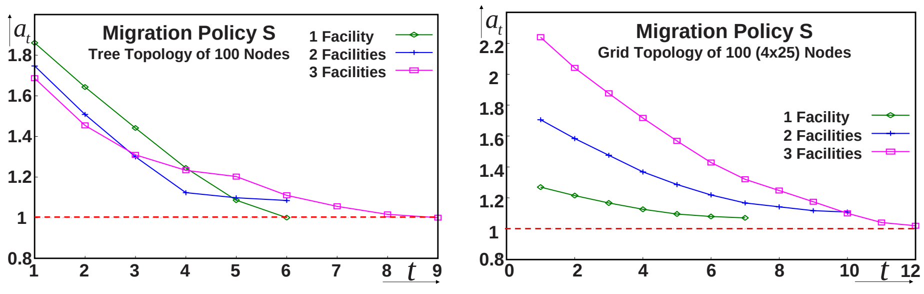
It should be mentioned that the end of a monotonically cost decreasing path is not necessarily the optimal position except for unique shortest path tree topologies and a single facility. In all other cases, a facility may stop away from the optimal position simply because there is a local minimum of the overall cost at that facility node. If the optimal position is at the end of the monotonically cost decreasing path, the facility will arrive at the optimal position and remain there (under Migration Policy S, E or H) as long as network conditions do not change.

VI. Simulation Results

A simulation tool was written in programming language C for creating network topologies (trees, grids, geometric random graphs [41], Erdős-Rényi random graphs [42], and Albert-Barabási graphs [43]) and implementing the migration policies. The goal of the simulations presented here is twofold: (a) to illustrate the behavior of the migration policies and show their accordance with the analytical study; (b) to capture cases not analytically studied (e.g., response to dynamic changes). For this reason, these results are not averaged over many experiments but rather the outcome of individual simulation experiments. Averaging would have given a macroscopic view (which is already covered by the analysis) but it would have missed the idiosyncrasies of the migration policies.

In all the results presented in this section, the initial position for the facilities is randomly chosen (and remains the same for all experiments concerning the same topology). Time t corresponds to facility movements and starts at time $t = 1$ in each case.

Figure 2 presents simulation results under Migration Policy S. In Figure 2.a (tree topology of 100 nodes and equal link weights), all three curves of a_t are monotonically decreasing. This is consistent with Theorem 1 and Theorem 2 (facilities are moving to neighbor nodes of smaller overall cost). When the cost cannot be further reduced, the facility stops moving. For a single facility, the approximation ratio becomes 1 (i.e., optimal position, also captured by Theorem 3) at time 6 (i.e., after 6 movements of the facility). For two facilities, facility movements stop at time 6; at this last movement $a_6 > 1$, implying that they did not arrive at the optimal positions (the analysis did not provide any guarantee for that). For three facilities, facility movements stop at time 9 where $a_9 = 1$, implying that the facilities arrived (and remained) at their optimal positions. Note that according to the analysis (Theorem



a.

b.

Fig. 2. Migration Policy S in tree and grid topologies of 100 nodes.

2), under Migration Policy S, if facilities do move, overall cost reduction is always achieved while they may or may not finally arrive at the optimal positions. In Figure 2.b (grid topologies of 100 nodes and equal link weights), as before, all facilities move along a monotonically cost decreasing path. However, they fail to arrive at the optimal positions ($a_t > 1$ in all cases). It is possible to arrive at the optimal positions but this is not guaranteed, as indicated by the analytical results.

Figure 3 presents simulation results under Migration Policies E and H. In Figure 3.a (Migration Policy E), for a single facility, $a_{30} = 1$. Note that $a_{24} = 1$, which means that the facility has moved to the optimal position as soon as time 24. However, it takes 6 more time units (i.e., time $t = 30$) for the facility to stop moving due to tentative movements under Migration Policy E. This overhead due to tentative facility movements is more obvious when compared to Migration Policy S (e.g., in Figure 2.b, facility movements stop at time 7 instead of 30). For two or three facilities, it is interesting to note that under Migration Policy E, the overall cost does not necessarily decrease (e.g., around time 28, the approximation ratio increases and decreases in both cases). Recall that according to the analysis, a monotonically cost decreasing path is followed under Migration Policy E for a single facility and topologies of equal link weights.

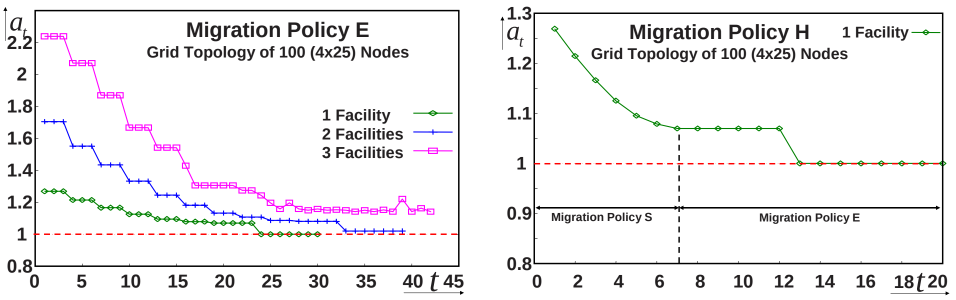
Figure 3.b presents results under Migration Policy H for a single facility. This case is better suited for Migration Policy E and, consequently, Migration Policy H. Note that $a_7 \approx 1.1$ is achieved as a result of Migration Policy S (see the resemblance of the curve depicted in Figure 2.b for a single facility). After some additional time, tentative movements to neighbor nodes take place under Migration Policy E (which is resumed after the facility stopped moving under Migration Policy S) and the facility stops at the optimal position, which in this run occurred at $t = 20$. It is interesting to see that under Migration Policy H (Figure 3.b), the facility moves to the optimal position in less time ($a_{20} = 1$) than under Migration Policy E ($a_{30} = 1$). This considerable saving in time (or number of movements) is due to the initial employment of Migration Policy S, which requires no tentative facility movements. Therefore, under Migration Policy H, both time and facility movements are saved when compared to Migration Policy E.

Tree and grid topolgies served well so far the purposes of demonstrating the particulars of the proposed migration policies and confirming the analytical results. In the sequel, migration is studied considering more *realistic topologies* like geometric random graphs (suitable for studying mobile ad hoc networks [41]), Erdős-Rényi random graphs (suitable for comparison reasons [42]), and Albert-Barabási graphs (power-law graphs that model many modern networks including the Internet [43]), in dynamically changing environments. More specifically, geometric random graphs are created considering a connectivity radious $r_c = 0.21$ around each node in the square plane $[0, 1] \times [0, 1]$, Erdős-Rényi random graphs considering connectivity probability $p_c = 0.1$ and Albert-Barabási based on preferential attachment [43].

Figure 4.a presents simulation results under Migration Policy S for various network sizes. Link weights are equal, which is a worst-case scenario under Migration Policy S. Let T denote *termination time* and let a_T correspond to a_t at termination time. Based on Figure 4.a, it is observed that the approximation ratio is not affected as the network size increases. It is also interesting to observe that the approximation ratio remains below 1.5, which is small considering the fact that equal link topologies do not allow for significant improvements under Migration Policy S. When Migration Policy E is employed for topologies such as those depicted in Figure 4.a, the approximation ratio is improved. However, the price to pay is the increased number of tentative facility movements and eventually the increased time delay (as depicted in Figure 4.b). It is obvious that termination time T increases as N increases for the realistic network topologies considered here, and therefore, prohibitive for the use of either Migration Policy E or H for large values of N.

The response to dynamically changing network environments is also investigated. In order to capture this effect using simulations, the previously described topologies are once again considered in an environment in which service demands change every time unit for a fraction of the network nodes, thus forcing changes in the optimal facility positions. Let β be the fraction of the network nodes that at each time unit keep unchanged their service demands. The rest (i.e., $(1 - \beta)N$) of the nodes set their own service demands to 1.0 (i.e., heavy service demands).

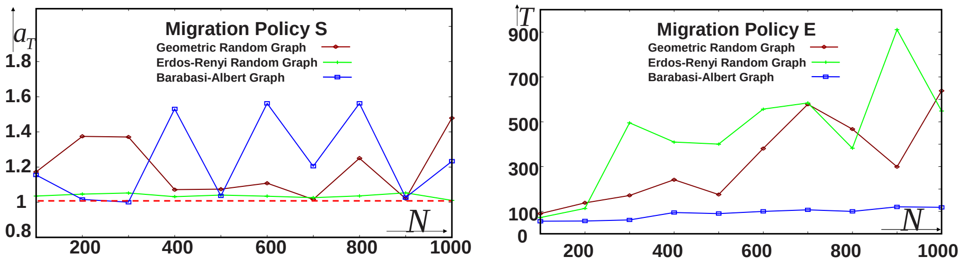
Figure 5 presents simulation results under Migration Policy S (the approximation ratio as a function of β). $\overline{a}_t$ is the averaged value of the approximation ratio a_t over time, without taking into account the values corresponding to the initial convergence period in each case. Figure 5.a and Figure



a.

b.

Fig. 3. Migration Policies E and H in grid topologies of 100 nodes.



a.

b.

Fig. 4. Approximation ratio and termination time as a function of N.

5.b illustrate that $\overline{a}_t$ does not dramatically change despite β. This is basically attributed to the fact that under Migration Policy S, facilities move immediately (i.e., without the need for any tentative facility movement) to more effective positions. If Migration Policy E (or H) was used, the need for tentative movements would have been prohibitive because by construction, this policy assumes that the network environment remains static as long as tentative facility movements take place. Given that Migration Policy E (and H) can be used only for one service facility and topologies of equal link weights, it is obvious that this policy is applicable to a narrow range of network environments, whereas Migration Policy S is applicable to all of network environments.

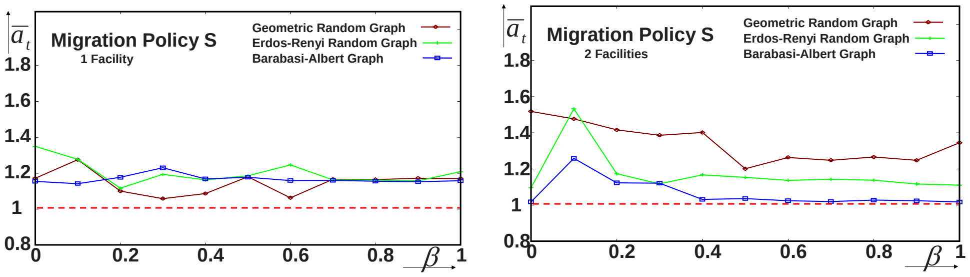
By comparing the results depicted in Figure 5.a and Figure 5.b, it is also interesting to observe that for two facilities: (a) for geometric random graphs, a_t slightly increases; (b) for Erdős-Rényi random graphs, a_t remains about the same; and (c) for Albert-Barabási random graphs, a_t decreases. The obvious conclusion is that depending on the topology, an increment of the number of facilities may eventually result in different values for the approximation ratio under migration.

In summary, in any (connected) topology and for any number of facilities moving under Migration Policy S, cost reduction is achieved. For topologies with equal link weights yielding multiple shortest path trees, a single facility and a static environment, Migration Policy H is capable of improving on the performance of Migration Policy S, by invoking Migration Policy E when the former cannot yield any further improvement. In addition, Migration Policy H is faster than Migration Policy E. In realistic topologies, it becomes obvious that as the network size increases, the overhead due to the tentative facility movements under Migration Policies E and H becomes prohibitive and Migration Policy S becomes the obvious choice. Moreover, in response to dynamic network changes, Migration Policy S is more effective due to its capability to move facilities to more effective positions almost immediately.

VII. Conclusions

Service migration was explored in this paper as a way of addressing the service placement problem in large scale and dynamic (autonomic) networking environments. Unlike classical approaches that require the knowledge of global network topology and detailed service demands, the proposed approach requires only local topology information (neighbors of the service hosting node) and aggregate service demands that become readily available to the service hosting node. Furthermore, as the network conditions change, one of the three migration policies presented here (Migration Policy S) inherently incorporates these changes in determining the migration path to follow in each step, as opposed to requiring to take a snapshot of the environment (network topology and service demands) at a certain time and to solve the corresponding large optimization problem (classical approach).

Regarding future work, as new network environments appear (e.g., social networking) and more services emerge, numerous facilities belonging to different services may eventually reside in a small number of (cost-effective) nodes resulting in problems like denial of service, single point of failure, energy consumption and fairness. In order to deal with these emerging environments, service migration should be further enhanced with the appropriate mechanisms to avoid concentration of numerous facilities of different services at the same nodes. Furthermore, future work should focus on real



a.

b.

Fig. 5. Averaged approximation ratio $\overline{a}_t$ under Migration Policy S as a function of β.

topology and service demands scenarios (e.g., input from real traces) and on the implementation of a testbed to be able to evaluate further on the applicability of the proposed policies.

References

[1] P.B. Mirchandani and R.L. Francis, “Discrete Location Theory,” John Wiley and Sons, 1990.

[2] O. Kariv, and S.L. Hakimi, “An algorithmic approach to network location problems, II: The p-medians,” SIAM Journal on Applied Mathematics, 37, 3 (1979), 539-560.

[3] K. Oikonomou, and I. Stavrakakis, “Scalable Service Migration: The Tree Topology Case,” The Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2006), Lipari, Italy, June 14-17, 2006.

[4] K. Oikonomou, I. Stavrakakis, and A. Xydias, “Scalable Service Migration in General Topologies,” The Second International IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2008), Newport Beach, California, 23 June, 2008.

[5] A. Tamir, “An O(pn) algorithm for p-median and related problems on tree graphs,” Operations Research Letters, 19:59–64, 1996.

[6] G. Wittenburg, and J. Schiller, “A Survey of Current Directions in Service Placement in Mobile Ad-hoc Networks,” Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom ’08), 2008, Hong Kong, 17-21 March 2008.

[7] D. S. Hochbaum, “Heuristics for the fixed cost median problem,” Mathematica Programming, 22:148–162, 1982.

[8] S. Guha, and S. Khuller, “Greedy strikes back: improved facility location algorithms,” J. Algorithms 31 (1999) 228-248.

[9] K. Jain, M. Mahdian, and A. Saberi, “A new greedy approach for facility location problems,” In Proc. ACM Symposium on the Theory of Computing (STOC’02), 2002.

[10] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani, “Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP,” Journal of the ACM, 2003.

[11] D. B. Shmoys, E. Tardos,, and K. Aardal, “Approximation algorithms for facility location problems,” In Proceedings of the 29th ACM Symposium on Theory of Computing, pages 265–274, 1997.

[12] F. Chudak and D. Shmoys. Improved approximation algorithms for capacitated facility location problem. In Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 875–876, 1999.

[13] M. Sviridenko, “An improved approximation algorithm for the metric uncapacitated facility location problem,” In W.J. Cook and A.S. Schulz, editors, Integer Programming and Combinatorial Optimization, Volume 2337 of Lecture Notes in Computer Science, pages 240-257, Springer, Berlin, 2002.

[14] K. Jain and V. V. Vazirani, “Primal-dual approximation algorithms for metric facility location and k-median problems,” in Proc of IEEE FOCS 99, New York City, NY, USA, 1999.

[15] K. Jain and V. Vazirani, “Approximation algorithms for metric facility location and kmedian problems using the primal-dual schema and Lagrangian relaxation,” Journal of ACM, 48(2):274–296, 2001.

[16] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman, “Analysis of a local search heuristic for facility location problems,” Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms, 1-10, 1998.

[17] V. Arya, N. Garg, R. Khandekar, K. Munagala, A. Meyerson, 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), pages 21–29. ACM Press, 2001.

[18] J.-H. Lin, and J.S. Vitter, “Approximation Algorithms for Geometric Median Problems,” Information Processing Letters, 44:245-249, 1992.

[19] M. Charikar, S. Guha, D. B. Shmoys, and E. Tardos, “A constant factor approximation algorithm for the k-median problem,” in Proc. of ACM STOC 99, Atlanta, GA, USA, 1999, pp. 110.

[20] I.D. 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), pages 661–670, January 2001.

[21] M. Mahdian, Y. Ye, and J. Zhang, “Improved Approximation Algorithms for Metric Facility Location Problems,” 5th International Workshop on Approximation Algorithms for Combinatorial Optimization, 2002.

[22] M. Mahdian and M. Pal, “Universal facility location,” in Proc. of ESA 03, Budapest, Hungary, 2003, pp. 409-421.

[23] N. Garg, R. Khandekar, and V. Pandit, “Improved approximation for universal facility location,” in Proc of ACM-SIAM SODA 05, 2005, pp. 959-960.

[24] T. Moscibroda, and R. Wattenhofer, “Facility Location: Distributed Approximation,” in 24th ACM Symp. on the Principles of Distributed Computing, Las Vegas, USA, July 2005.

[25] D. Krivitski, A. Schuster, and R. Wolff, “A Local Facility Location Algorithm for Large-Scale Distributed Systems, Journal of Grid Computing, Volume 5, Number 4 / December, 2007, pp. 361–378.

[26] C. Ragusa, A. Liotta, G. Pavlou, “An Adaptive Clustering Approach for the Management of Dynamic Systems,” IEEE Journal of Selected Areas in Communications (JSAC), special issue on Autonomic Communication Systems, Vol. 23, No. 12, pp. 2223–2235, IEEE, December 2005.

[27] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, and A. Bestavros, “Distributed Placement of Service Facilities in Large-Scale Networks,” IEEE INFOCOM’07, May 6-12, 2007, Anchorage, Alaska.

[28] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman, “Placement algorithms for hierarchical cooperative caching,” In Proceedings of the 10th Annual Symposium on Discrete Algorithms (ACM-SIAM SODA), pages 586-595, 1999.

[29] L. Yamamoto, and G. Leduc, “Autonomous reflectors over active networks: towards seamless group communication,” AISB journal, special issue on agent technology, 1(1):125–146, December 2001.

[30] A. Meyerson, “Online facility location,” in FOCS 01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science, Washington, DC, USA, 2001.

[31] B. Li, and K. H. Wang, “NonStop: Continuous Multimedia Streaming in Wireless Ad Hoc Networks with Node Mobility,” IEEE Journal on Selected Areas in Communications, 21(10):1627-1641, Dec. 2003.

[32] N. Laoutaris, V. Zissimopoulos, and I. Stavrakakis, “Joint Object Placement and Node Dimensioning for Internet Content Distribution,” Information Processing Letters, Vol. 89, No. 6, pp. 273-279, March 2004.

[33] K. Y. K. Liu, J. C. S. Lui, and Z.-L. Zhang, “Distributed Algorithm for Service Replication in Service Overlay Network,” Third IFIP-TC6 Networking Conference (Networking 2004), pages 1156–1167, 2004.

[34] F. Sailhan, and V. Issarny, “Scalable Service Discovery for MANET,” 3rd IEEE Intl. Conf. on Pervasive Computing and Communications, Kauai, USA, Mar. 2005.

[35] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, “Gateway Placement Optimization in Wireless Mesh Networks With QoS Constraints,” IEEE Journal on Selected Areas in Communications, 24(11):2127-2136, Nov. 2007.

[36] D. Krivitski, A. Schuster, R. Wolff, “A local facility location algorithm for sensor networks,” Proceedings of the International Conference on

Distributed Computing in Sensor Systems (DCOSS05), Marina del Rey, CA, USA, June 2005.

[37] C. Frank, and K. Römer, “Distributed Facility Location Algorithms for Flexible Configuration of Wireless Sensor Networks,” In 3rd IEEE Intl. Conf. on Distributed Computing in Sensor Systems, Santa Fe, USA, June 2007.

[38] D. Oppenheimer, B. Chun, D. Patterson, A. C. Snoeren, and A. Vahdat, “Service placement in a shared wide-area platform,” Proc. of USENIX06, 2006, pp. 2626.

[39] T. Loukopoulos, P. Lampsas, and I. Ahmad, “Continuous replica placement schemes in distributed systems,” Proc. of ACM ICS05, Boston, MA, 2005.

[40] D. Bertsekas, and R. Gallager, “Data networks,” 2nd edition, Prentice-Hall, Inc., 1992.

[41] M. Penrose, “Random geometric graphs,” Oxford Univarsity Press Inc., 2003.

[42] B. Bollobás, “Random Graphs,” Cambridge University Press, Second Edition, 1999.

[43] R. Albert and A. Barabási, “Statistical mechanics of complex networks,” Rev. Mod. Phys., 2002.

Appendix

A. Proof of Lemma 1

With respect to some node y, the distance from any node v over any shortest path of T^y_{t+1}, is smaller than or equal to the distance over a shortest path of any other shortest path tree of different root (e.g., T^x_t). Therefore, $d(v,y) \leq d^{T^x_t}(v,y)$, $\forall v \in V$, and based on Equation (1), $C_{t+1}(y) \leq C^{T^x_t}_{t+1}(y)$. The equality holds for the particular case that $T^y_{t+1} = T^x_t$, as it is the case for unique shortest path tree topologies. For any node $v \in I^y(T^x_t)$, $d^{T^x_t}(v,y) = d(v,x) - w(x,y)$, while for any node $v \in T^x_t \setminus I^y(T^x_t)$, $d^{T^x_t}(v,y) = d(v,x) + w(x,y)$. From the above and in view of Equations (1) and (2), it is derived that, $C^{T^x_t}_{t+1}(y) - C_t(x) = -\sum_{\forall v \in I^y(T^x_t)} \lambda_v w(x,y) + \sum_{\forall v \in T^x_t \setminus I^y(T^x_t)} \lambda_v w(x,y) = (\Lambda(T^x_t \setminus I^y(T^x_t)) - \Lambda(I^y(T^x_t)))\, w(x,y)$.

B. Proof of Theorem 1

Based on Lemma 1 and the fact that $w(x,y) > 0$, when $\Lambda(T^x_t \setminus I^y(T^x_t)) < \Lambda(I^y(T^x_t)$, it is ensured that $C^{T^x_t}_{t+1}(y) < C_t(x)$. Since $C^{T^x_t}_{t+1}(y) \geq C_{t+1}(y)$, $C_{t+1}(y) < C_t(x)$ is also satisfied. Given that there is a single facility in the network, $C_{t+1} < C_t$ is also satisfied.

C. Proof of Theorem 2

Let $f_t(v)$ denote the facility node corresponding to some network node v at time t. Let Z be a set of nodes such that, $Z = \{z : \forall z \in V \text{ and } f_{t+1}(v) \neq f_t(v)\}$. The case of $Z = \emptyset$ is trivial to prove based on Theorem 1. For the case that $Z \neq \emptyset$, let C'_{t+1} represent a *hypothetical overall cost* at time $t+1$, assuming all nodes $v \in Z$ forced to continue to be served by the same facilities as it was the case at time t (i.e., K_t instead of K_{t+1}). Based on the definition of the hypothetical cost $C^{T^x_t}_{t+1}(y)$ in Section IV, $C'_{t+1} = \sum_{\forall x \in K_t} C^{T^x_t}_{t+1}(y)$. For all facility movements $x \xrightarrow{t} y$, $x \in K_t$ and $y \in K_{t+1}$, under Migration Policy S, $C_t(x) > C^{T^x_t}_{t+1}(y)$ is satisfied, or, by summing up for all facility nodes, $C_t > \sum_{\forall x \in K_t} C^{T^x_t}_{t+1}(y) = C'_{t+1}$. Then, $C_{t+1} = C'_{t+1} + \Delta$, where Δ is the cost difference contributed by those nodes $v \in Z$ (cost contributed using the new facilities minus the cost contributed using the previous facilities), or, $\Delta = \sum_{\forall v \in Z} \lambda_v d(v, f_{t+1}(v)) - \sum_{\forall v \in Z} \lambda_v d(v, f_t(v))$. However, $d(v, f_{t+1}(v)) < d(v, f_t(v))$, (smaller distances force nodes $v \in Z$ to utilize new facilities). Therefore, $\Delta < 0$, and consequently, $C_{t+1} < C'_{t+1}$. Eventually, $C_{t+1} < C_t$.

D. Proof of Theorem 3

The cost corresponding to the facility located at some node v and the facility located at the optimal position (e.g., node u), can be shown as monotonically increasing by the number of hops away from the optimal position in a topology of a unique shortest path tree, [3]. This is due to the fact that the aggregate service demands – corresponding to the particular neighbor (of the facility) node that is towards the optimal position – monotonically increase. Since there is a unique shortest path tree, the facility will always move to nodes of lower overall cost (Theorem 1).

E. Proof of Lemma 3

Since $w(x,y) = 1$, it appears that for any node $v \in \Phi_{x \xrightarrow{t} y}$, $d(v,y) - d(v,x) < 1$. Note that when the service is located at node x at time t, $d(v,y) - d(v,x) \geq -1$. Given that $d(v,y)$ and $d(v,x)$ take discrete values, both previous inequalities are satisfied for, $d(v,y) - d(v,x) = 0$, or, $d(v,y) - d(v,x) = 1$.

F. Proof of Lemma 4

Based on the migration rule, $\Phi_{y \xrightarrow{t+1} x} \cap \Phi^{-1}_{x \xrightarrow{t} y} = \emptyset$. The next step is to show that there is no node $v \in \Phi_{x \xrightarrow{t+2} y}$, such that $d(v,y) - d(v,x) = -1$. Assume that there is a node v such that $v \in \Phi_{x \xrightarrow{t+2} y}$, and $d(v,y) - d(v,x) = -1$. Since this is valid for the facility movement $x \xrightarrow{t+2} y$, it should have been valid for $x \xrightarrow{t} y$ as well. Given that, $\Phi_{y \xrightarrow{t+1} x} \cap \Phi^{-1}_{x \xrightarrow{t} y} = \emptyset$, when the facility returns to node x, $v \in I^y(T^x_{t+2})$. Since $v \notin T^x_{t+2} \setminus I^y(T^x_{t+2})$, then $v \notin \Phi_{x \xrightarrow{t+2} y}$, which is a contradiction. Eventually, $\Phi^{-1}_{x \xrightarrow{t+2} y} = \emptyset$.

G. Proof of Lemma 5

For a facility movement $x \xrightarrow{t} y$ under Migration Policy S, it is enough to prove that under Migration Policy E, the facility will also move to node y at time $t+3$ (taking into account the tentative movements). Under Migration Policy S, $\Lambda\big(T^x_t \setminus I^y(T^x_t)\big) < \Lambda\big(I^y(T^x_t)\big)$ is satisfied. It is enough to show that the condition of Migration Policy E (i.e., $\Lambda\big(I^x(T^y_{t+1})\big) < \Lambda\big(I^y(T^x_{t+2})\big)$) is also satisfied. Due to the migration rule, $I^y(T^x_t) \subset I^y(T^x_{t+2})$, or, $\Lambda\big(I^y(T^x_t)\big) < \Lambda\big(I^y(T^x_{t+2})\big)$. Given that $T^x_t \setminus I^y(T^x_t) = I^x(T^y_{t+1}) \cup \Phi_{x \xrightarrow{t} y}$, $I^x(T^y_{t+1}) \subseteq T^x_t \setminus I^y(T^x_t)$ (the equality holds for those cases that $\Phi_{x \xrightarrow{t} y} = \emptyset$). Therefore, $\Lambda\big(I^x(T^y_{t+1})\big) \leq \Lambda\big(T^x_t \setminus I^y(T^x_t)\big)$, is satisfied. Since $\Lambda\big(T^x_t \setminus I^y(T^x_t)\big) < \Lambda\big(I^y(T^x_t)\big)$, $\Lambda\big(I^x(T^y_{t+1})\big) < \Lambda\big(I^y(T^x_{t+2})\big)$ is also satisfied. The reverse of the lemma’s hypothesis is true for those cases that $\Phi_{x \xrightarrow{t} y} \neq \emptyset$.