Scalable Service Migration: The Tree Topology Case

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

***Abstract*— Scalable *service placement* is a challenging problem in dynamic environments such as ad hoc or autonomic networks. Existing approaches typically consider static and reduced size networking environments and try to determine the *optimal service position* (the node at which some cost is minimized) by solving the 1-*median problem*. Since such approaches are complex and are based on global information knowledge, they are non-scalable and cannot cope with dynamic environments. A more reasonable approach to service placement for large, ad hoc and autonomic environments would be through *service migration*. That is, instead of solving continuously a large optimization problem requiring global information, consider policies for *moving* the service position (one hop/node at a time) based on *local* information, *towards* more effective positions. Developing service migration policies with good properties is a major challenge, since such policies may be sub-optimal (that is, they never converge to the optimal position), follow a non-monotonically cost decreasing path to the optimal position, etc.**

In this paper, the aforementioned issues are discussed and a simple service migration policy is proposed for undirected tree topologies. For this case it is shown analytically that the information available at the *current* service node only is sufficient for determining the direction towards nodes with monotonically decreasing service provision costs. Consequently, the proposed policy moves the service continuously towards the optimal position *in every step* and reaches the optimal position through a *shortest path* migration trajectory. As the optimal position may change in a dynamic environment, the proposed policy adapts the service migration path continuously towards the currently optimal position. Although some of the main results are also applicable to general network topologies, future work will focus on the general network topology by borrowing ideas from the current work.

***Index Terms*— Service, placement, migration**

I. INTRODUCTION

In traditional networks service provision is typically the responsibility of the (sub-) network owner or a well-defined entity that owns or leases part of the needed infrastructure and sometimes enters in agreement with network service providers. In such networks the location of the service provision is dictated by ownership limitations. The globalization of the Internet and the expansion of the service demand profiles have necessitated the careful selection of the location of the service, as well as the replication of the service provision points. The objective being, 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. The problem of service placement has received some attention in the aforementioned traditional networking environment, for example, in the context of content placement and replication in Content Distribution Networks, [1]. This problem is typically addressed by invoking approaches that do not scale with the number of services and network nodes, typically rely on some global information knowledge in order to provide for a solution under given (static) conditions and cannot inherently cope with dynamic environments. As indicated in the next paragraphs, the service generation and provision landscape and the supporting networking infrastructure are changing drastically in a way that the traditional approach to service placement is non-scalable.

The first change has to do with the proliferation and "miniaturization" of the services produced by networked nodes. The emerging "long-tail" relation between percentage of content (service) produced by a certain content (service) producer reveals the fact that network services proliferate in number and type and that most of these services are "small" (i.e., easily produced by small networked nodes). In addition, the technology appears to be mature to consider service personalization and autonomic service composition, [2], [3], which is expected to further enhance the "miniaturization" and proliferation of the network services.

The second change has to do with the proliferation and "miniaturization" or the network elements, as well as network users. The term "miniaturization" may be used here to capture the fact that the traditionally heavy network elements (routers) are increasingly being supplemented by lighter network elements that are contributed by (until recently) traditional network users; these users are becoming powerful enough to engage

This work has been supported in part by the project Autonomic Network Architecture (ANA) (IST-27489), which is funded by IST FET Program of the European Commission.

Konstantinos Oikonomou and Ioannis Stavrakakis are with the Department of Informatics and Telecommunications. E-mail: {okon, ioannis}@di.uoa.gr.

in ad hoc networking and contribute to the networking infrastructure. For example, the increasing contribution to the networking infrastructure of numerous small (traditionally) user-nodes is already materializing in last-mile networking and is expected to dominate soon (e.g., home owner based WLAN network service access). In addition, numerous new small appliances are increasingly being networked, contributing to the proliferation and miniaturization of the network users. The increasing proliferation of network infrastructure and its increasing autonomicity and ownership diversification are the main reasons for which a new network architecture is needed to organize and address the efficiency and complexity issues associated with it, [4].

The main point from the above discussion is the proliferation of services and network nodes that calls for approaches that scale well with the numbers; "miniaturization" contributes to the proliferation, as well as calls for approaches that should be distributed and relatively light. Consequently, the traditional problem of placing relatively few big services to one of the few (powerful) potential service provider facilities (big network elements) is increasingly being transformed 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), in the sense that some average cost associated with the provision of this service is minimized. When p service nodes are considered, the optimal placement of these p service nodes can be determined by formulating and solving a *p-median problem*, [5]. The p-median problem has been shown to be an NP-hard problem for general graphs, [6]. For the special case of the 1-median problem and an undirected tree topology (considered in this paper), the number of exchanged messages remains as high as $O(N^2)$, [6], [7], where N is the number of nodes in the network. In most cases, apart from some heuristic or approximation policies presented later in Section II, the determination of the optimal service node requires some *global information* of the *network status* (e.g., the network topology and the service demands of all network nodes).

In addition to the aforementioned complexity and scalability issues associated with the solution to the 1-median problem, such approaches become prohibitive in dynamic environments where a repeated application of the approach and continuous dissemination of global status information would be required, to continuously determine the updated optimal service node location. For such dynamic environments, one could either incur the cost of determining the optimal position continuously at predefined periodic intervals, or implement (typically complex) mechanisms for detecting when the optimal service node position is not valid any more and determine the new optimal service node position. In any case, the cost is heavy and the effectiveness of these approaches questionable (i.e., likelihood to be actually providing the service when requested from the optimal location may not be high).

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*. That is, instead of solving continuously a large optimization problem requiring global information, consider policies for *moving* the service position (one hop/node at a time) based on *local* information, *towards* more effective positions, [8]. Developing service migration policies with good properties is a major challenge, since such policies may be sub-optimal (that is, they never converge to the optimal position), follow a non-monotonically cost decreasing path to the optimal position, etc.

In this paper, a simple migration policy is proposed for undirected tree topologies. It is shown analytically that the information available at the *current* service node only is sufficient for determining the direction towards nodes with monotonically decreasing cost provision costs, and eventually, the optimal service node.

The service node needs to simply monitor the *aggregate* amount of data exchanged through its neighbor nodes associated with the particular service, and decide on the service movement based *exclusively* on the information gathered through the monitoring process. An important result derived in this paper shows that this information is *adequate* in order for the service to move towards the optimal service node. It is also proved that under the proposed policy the service is finally moved to the optimal service node and remains there as long as the network status does not change significantly; minor changes typically do not result in a new optimal service node position). When major changes occur, the service eventually moves towards the new optimal service node position.

Based on analytic studies it is shown that the proposed policy: (a) Moves the service to the optimal position when there are no changes to the network status and service demand profiles (i.e., it is efficient in a static environment); (b) Adapts dynamically to changes to the network status and service demand profiles (i.e., it is efficient in a dynamic environment); (c) It is scalable; (d) It has low complexity (no additional message exchanges).

Consequently, the proposed policy presents better characteristics than existing (and static) approaches in the area (e.g. [6], [7]).

Section II overviews the p-median problem and presents past related work in the area. Section III presents a detailed description of the network model used throughout this paper. The key contributions of this paper are included in Section IV. A theorem presented therein shows that the difference in the servise provision cost when the service node is placed in two neighbour nodes does not depend on the weights (costs for using) of all the links in the network but only on the weight of the link among them; in addition, knowledge of the aggregate service demands at a particular node is sufficient to determine whether there exist a neighbor node with lower service provision cost. The aforementioned observations serve as the motivation behind the service migration policy proposed in Section IV as well. Analytical results are presented in Section V. It is shown that the direction of the movement is unique and that it follows a monotonically cost decreasing path towards the optimal service node position. Eventually, the service arrives at the optimal service node and remains there for as long as it remains the optimal one. When the network status changes, the service is able to move towards the new optimal location, irrespectively of its current position in the network. In Section VI some practical issues are considered. The conclusions are drawn in Section VII.

II. The p-median Problem

Suppose that 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 the edges of the graph be assigned a positive integer referred to as *weight*. Let $d_{u,v}$ denote the *distance* between node u and node v, corresponding to the summation of the weights along a *shortest path* among the two nodes (for the same node $d_{v,v} = 0$). Alternatively, $d_{u,v}$ denotes the *traveling cost* between node u and node v.

Let λ_v denote the rate at which data packets are transferred through the network between node v and the service node for the particular service: λ_v will be referred to as the *service demands* of node v. Let X_p be the set of p nodes at which the service is located. That is, it is assumed that there are p nodes which are capable of providing a given service. For a given placement X_p of these nodes and assuming that the cost of service provision is directly proportional to the amount of data transferred per unit time (λ_v) and the distance travelled ($d_{u,v}$), the total cost of service provision, $C(X_p)$, is given by, $\forall X_p \in \mathbb{X}_p$,

$$C(X_p) = \sum_{\forall v \in V} \lambda_v \min_{u \in X_p} \{d_{v,u} : u \in X_p\}, \qquad (1)$$

where $\mathbb{X}_p$ is the set of all possible p-*placements*, [5]. The solution of the p-*median problem* amounts to determining the placement X_p such that $C(X_p)$ is minimized. Figure 1 depicts a graphical representation of a 2-median problem (the service is located at nodes y and δ denoted by the dotted ellipses).

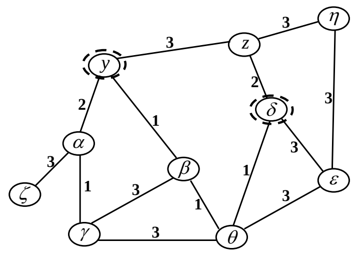
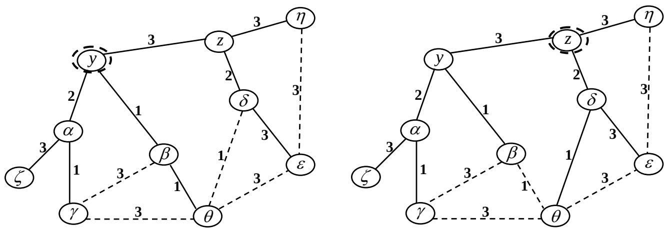


Fig. 1. Example of a 2-median problem for a general network topology.

It has been proved by Kariv and Hikimi, [6], that the p-median problem is NP-hard for the general directed graph. For the case of undirected trees they have also proved that the problem has $O(p^2N^2)$ complexity; Tamir, [7], more recently has reduced this complexity to $O(pN^2)$. There is also considerable work in the area of approximation algorithms and heuristics, [9], [10], [11], [12], that try to reduce further the complexity of the problem. More recent work in the area can be found in [13], [1], however, a complete list of past work on the p-median problem is hard to compile since this has been an active research area for many decades with applications in communication networks, engineering, computer science, economics, etc. The interested reader is advised to have a look at reference [5] for further past related work.

III. System Description

In every communication network it is the responsibility of the employed *routing protocol* to provide for the efficient forwarding of the data packets. In this paper, it will be assumed that the routing protocol is capable of delivering data packets along a shortest path. Consequently, for a certain destination node, a *minimum spanning tree* can be constructed rooted at the particular node, [14]. An example is depicted graphically in Figure 2, with respect to the network of Figure 1. Assuming that there is only one service in the network (1-median problem) located at node y, it is evident that various data packets from the nodes will arrive at node y along the shortest path (indicated in Figure 2.a as dense lines). It can be observed that a tree topology is created (dense lines). The same applies for the case of node z as the service node depicted in Figure 2.b. However, for this case the corresponding minimum spanning tree is different than that depicted in Figure 2.a (e.g. the link among node θ and node δ).



a. Service at node y. b. Service at node z.

Fig. 2. Data packets are forwarded in a network towards their destination along a shortest path.

From the previous discussion it is evident that the study of tree topologies is quite relevant in networking environments, [14]. Many of the results presented in this paper refer to such topologies and can be the basis for further investigations regarding more general network topologies. For the rest of this paper, all mentioned topologies will be undirected and symmetrical (when a direct link of some weight exists between node x and node y then a direct link of the same weight exists between node y and node x).

Let $tree_v^u$ denote the set of nodes of the *subtree below* node v (including node v) for some *ancestor* node u; node u is an ancestor of node v and node v is a *descendant* of node u. Clearly, $tree_v^{u_i} = tree_v^{u_j}$, for all nodes u_i, u_j, that are ancestors of node v; the exponent basically indicates that the particular node is *not* part of the subtree rooted at node v.

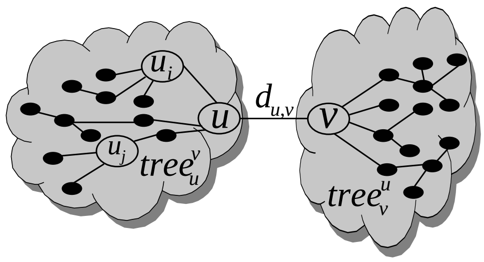


Fig. 3. Example for sets $tree_u^v$ and $tree_v^u$ when $u \in S_v$. For the former case, node v is an ancestor of node u, while for the latter case, node u is an ancestor of node v. Note that $tree_v^{u_i} = tree_v^{u_j} = tree_v^u$.

For the special case that node u and node v are neighbor nodes ($u \in S_v$),

$$V = tree_u^v \cup tree_v^u. \tag{2}$$

Notice that $tree_v^u$ does not imply that node u is necessarily a neighbor node of v. Node u is an ancestor of node v and not necessarily a *parent*, just like node v is a descendant of node u but not necessarily a *child*. The definition of the aforementioned $tree_v^u$ facilitates the description of the aggregate service demands generated from nodes v and below in the tree. Consider, for example, that u is the service node. The data packets will travel over the branches of a (minimum spanning) tree for which node u is the root. For this case, node v is a descendant of node u and all descendants of node v belong in $tree_v^u$. It is important to note that data packets exchanged between the service node u and any node $x \in tree_v^u$, will be forwarded through node v. Eventually, v is the node that forwards the aggregate amount of data for all nodes $x \in tree_v^u$ towards their destinations (either node u or node x). Given that λ_x corresponds to the service demands of node x, the *aggregate service demands* of those nodes belonging to $tree_v^u$, denoted by Λ_v^u, is given by,

$$\Lambda_v^u = \sum_{\forall x \in tree_v^u} \lambda_x. \tag{3}$$

An example case for a service node y is depicted in Figure 4.

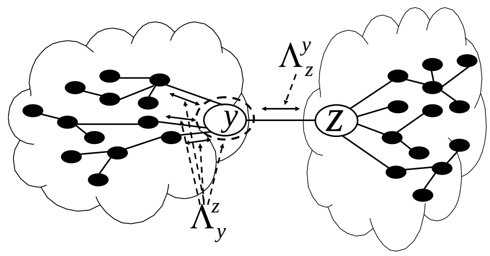


Fig. 4. The double dense arrows denote aggregate service demands for a particular link. The dotted arrows point to the links that their aggregate service demands correspond to Λ_y^z and Λ_z^y. Note that λ_y is included in Λ_y^z, since $y \in tree_y^z$.

In order to simplify the notations, for the particular case of the 1-median problem, Equation (1) can be written in the following form, where u denotes the node that is assumed to be the service node.

$$C_u = \sum_{\forall v \in V} \lambda_v d_{v,u}. \tag{4}$$

The solution to the resulting 1-median problem amounts to determining the node u that minimizes C_u. Let u_{opt} denote that node that minimizes the mean cost of servicing demands in V.

IV. The Proposed Policy

The service migration philosophy, as it was briefly described earlier in Section I, is based on a per hop movement of the service in the network. A simple policy, that is in line with this philosophy, is proposed in this section. The following theorem is useful.

Theorem 1: Considering nodes y and z as service nodes, with $z \in S_y$, the difference between the costs, C_y, and C_z, is given by,

$$C_z - C_y = \left(\Lambda_y^z - \Lambda_z^y\right) d_{y,z}. \quad (5)$$

Proof: Since nodes y and z are neighbors, $V = tree_z^y \cup tree_y^z$ (see Equation (2)). Consequently, the corresponding costs of locating the service at node z or y (see Equation (4)) are given by, $C_z = \sum_{\forall v \in V} \lambda_v d_{v,z} = \sum_{\forall v \in tree_z^y} \lambda_v d_{v,z} + \sum_{\forall v \in tree_y^z} \lambda_v d_{v,z}$ and $C_y = \sum_{\forall v \in V} \lambda_v d_{v,y} = \sum_{\forall v \in tree_y^z} \lambda_v d_{v,y} + \sum_{\forall v \in tree_z^y} \lambda_v d_{v,y}$.

Since, $d_{v,z} = d_{v,y} + d_{y,z}$, $\forall v \in tree_y^z$, C_z can be rewritten as follows, $C_z = \sum_{\forall v \in tree_z^y} \lambda_v d_{v,z} + \sum_{\forall v \in tree_y^z} \lambda_v d_{v,y} + \sum_{\forall v \in tree_y^z} \lambda_v d_{y,z}$. Since, $d_{v,y} = d_{v,z} + d_{y,z}$, $\forall v \in tree_z^y$, C_y can be rewritten as follows, $C_y = \sum_{\forall v \in tree_y^z} \lambda_v d_{v,y} + \sum_{\forall v \in tree_z^y} \lambda_v d_{v,z} + \sum_{\forall v \in tree_z^y} \lambda_v d_{y,z}$.

From the above and in view of Equation (3), it is easily derived that, $C_z - C_y = \sum_{\forall v \in tree_y^z} \lambda_v d_{y,z} - \sum_{\forall v \in tree_z^y} \lambda_v d_{y,z} = \Lambda_y^z d_{y,z} - \Lambda_z^y d_{y,z} = \left(\Lambda_y^z - \Lambda_z^y\right) d_{y,z}$. ■

In view of Equation (5) two interesting observations are possible regarding the difference of the cost when the service 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, $d_{y,z}$. Second, it depends on the *difference of the aggregate service demands*.

Consequently, it is evident from Equation (5) that 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 in order to determine differences in costs associated with neighboring service nodes and, eventually, determine the service node that induces the lowest cost among neighboring nodes. Even knowledge of $d_{y,z}$ is *not necessary*, as it is shown later in Lemma 1. What is actually required is information regarding the aggregate service demands (e.g., Λ_y^z, Λ_z^y) at the service node (e.g., node y). Note that the aggregate service demands at a service node, say Λ_y^z and Λ_z^y for service node y, can be estimated by employing a statistical monitoring process at node y. The estimation is facilitated by the fact that it is applied to aggregate service demands, as it is presented later in Section VI. An example regarding Λ_y^z and Λ_z^y is depicted in Figure 4.

Lemma 1: $C_z < C_y$ for $z \in S_y$, iff $\Lambda_y^z < \Lambda_z^y$.
The proof is trivial in view of Equation (5) and the fact that $d_{y,z} > 0$, when $y \neq z$. ■

The proposed Service Migration Policy: The service is moved from node y to the neighbor node z, $z \in S_y$, iff $\Lambda_y^z < \Lambda_z^y$.

According to the proposed policy and in view of Lemma 1, it is easy to conclude that every movement of the service results in cost reduction. In addition, the proposed policy is of low complexity in the sense that no message exchanges are required among the nodes in the network for the purpose of implementing the policy (it is exclusively based on information available at the service node).

It is important to emphasize at this point that the proposed policy is capable of achieving cost reduction for a *general network topology* and not only for the case of a tree that is considered here. In particular, for a general topology data packets are forwarded towards the service node y through a certain minimum spanning tree (determined by an underlying routing protocol) where node y is the root; let MST^y denote such a minimum spanning tree. If $z \in S_y$ in MST^y (i.e., node z is a neighbor of node y in MST^y) and $\Lambda_y^z < \Lambda_z^y$, then according to the proposed policy the service is moved to node z. In the sequel, it is shown that moving the service from node y to node z indeed reduces the cost in general topologies as well.

Suppose that the service is moved from node y to node z as indicated above. The cost of providing service from node z is shaped by the distances between the nodes of the network and node z, according to MST^z. If $MST^z \equiv MST^y$ (i.e., data packets are forwarded along the branches of the same tree no matter whether the service is located at node y or z), then Lemma 1 holds for this common tree and the cost is indeed reduced ($C_z < C_y$). As it is shown in [14], there always exists a unique minimum spanning tree as long as the weights of the links are distinct.

However, if $MST^z \neq MST^y$ (see, e.g., Figure 2), MST^z will contain by definition shorter or equal distances between the network nodes and node z, than in MST^y. Therefore, the actual cost of service from node z and according to MST^z ($\neq MST^y$), say C'_z, cannot exceed the cost of serving from node z and according to MST^y (C_z); i.e., $C'_z \leq C_z$. Since $C_z < C_y$ and $C'_z \leq C_z$, it is concluded that $C'_z < C_y$ when $MST^z \neq MST^y$. Thus, the proposed policy indeed moves the service towards lower costs in a general network topology.

In view of the above discussion, it is concluded that for general topologies the proposed policy moves the service to a neighbor node that can provide the service at a lower cost. While the selected neighbor node is the one with the lowest service provision cost among all neighbors in the undirected tree topologies, thus may not be the case in general network topologies. In general network topologies lower costs may be actually achieved by a new service node if the associated minimum spanning tree differs from that of the original service node.

V. Analysis

It is important to prove that, when the proposed policy is employed, the service (a) is finally located at the optimal service node, thus achieving cost minimization and not only cost reduction; (b) stays at the optimal service node as long as the network status does not change. The latter is shown in the following lemma.

Lemma 2: If the service is located at the optimal service node, then it is not moved to any other node.
The proof is trivial considering Lemma 1 and the fact that $C_z > C_y$, $\forall y \in S_z$. ∎

In order to prove that in a static environment the service arrives at the optimal service node, it is important to prove previously the *uniqueness* of the direction of the service movement.

Theorem 2: For any service node there is at most one neighbor node z, $z \in S_y$, such that $C_y > C_z$.

Proof: If node z is the only neighbor of the service node y ($|S_y| = 1$), the proof is trivial.

Consider now the case where $|S_y| \geq 2$. Let node $z \in S_y$ be such that $C_z < C_y$. Let node $\theta \in S_y - \{z\}$. It is sufficient to show that $C_y < C_\theta$. Let K be the set of nodes, $K = S_y - \{z\} - \{\theta\}$. Obviously, $|K| \geq 0$, depending on the number of neighbor nodes of node y.

From Figure 5, it is possible to observe that $tree_y^z = tree_\theta^z \cup_{\forall k \in K} tree_k^z \cup \{y\}$ and $tree_\theta^z = tree_\theta^y$. From the previous and according to Equation (3), $\Lambda_y^z = \Lambda_\theta^y + \sum_{\forall k \in K} \Lambda_k^z + \lambda_y$. The same way, from Figure 5, it is possible to observe that $tree_y^\theta = tree_z^\theta \cup_{\forall k \in K} tree_k^z \cup \{y\}$ and $tree_z^\theta = tree_z^y$. Accordingly, $\Lambda_y^\theta = \Lambda_z^y + \sum_{\forall k \in K} \Lambda_k^z + \lambda_y$. By summing up the former expressions for Λ_y^z and Λ_y^θ, the following is obtained: $\Lambda_y^z + \Lambda_y^\theta = \Lambda_\theta^y + 2\sum_{\forall k \in K} \Lambda_k^z + 2\lambda_y + \Lambda_z^y$, or $\Lambda_y^z - \Lambda_z^y = \Lambda_\theta^y - \Lambda_y^\theta + 2\left(\sum_{\forall k \in K} \Lambda_k^z + \lambda_y\right)$.

Since, $C_y > C_z$ it is concluded from Lemma 1 that $\Lambda_y^z < \Lambda_z^y$, or $\Lambda_y^z - \Lambda_z^y < 0$. Consequently, $\Lambda_\theta^y - \Lambda_y^\theta + 2\left(\sum_{\forall k \in K} \Lambda_k^z + \lambda_y\right) < 0$, or $\Lambda_\theta^y + 2\left(\sum_{\forall k \in K} \Lambda_k^z + \lambda_y\right) < \Lambda_y^\theta$. Given that $\sum_{\forall k \in K} \Lambda_k^z + \lambda_y \geq 0$, $\Lambda_\theta^y < \Lambda_y^\theta$. Finally, from Lemma 1, it is concluded that $C_\theta > C_y$ and the proof is completed. ∎

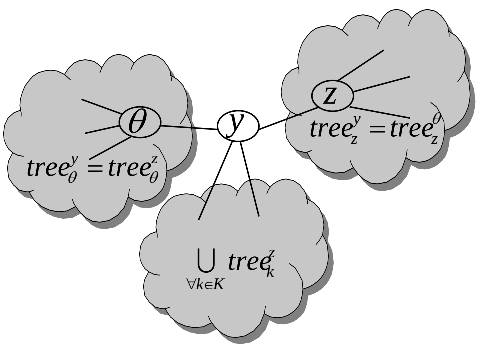


Fig. 5. Sets $tree_\theta^y$, $tree_z^y$ and $tree_k^z$, $\forall k \in K$.

Theorem 2 is useful for the next step. That is, to prove that irrespectively of the current service node, the service will, eventually, arrive at the optimal service node. Lemma 2 ensures the fact that it will stop moving at the optimal service node.

Theorem 3: The service eventually arrives at the optimal service node, when it is moved in the network according to the proposed Service Migration Policy.

Proof: It is sufficient to show that the service will move along a path of monotonically decreasing cost and that the node at the end of the path is the optimal service node.

Suppose that a particular node v is located H hops away from the optimal service node u_{opt}. Let h_0 denote node u_{opt}, h_1 the neighbor node of u_{opt} along the path towards node v, h_2 the next one, ..., until node h_H that is identical to node v (see also Figure 6).

According to Equation (5),

$$\begin{aligned}
C_{h_1} - C_{h_0} &= \left(\Lambda_{h_0}^{h_1} - \Lambda_{h_1}^{h_0}\right) d_{h_0,h_1} \\
C_{h_2} - C_{h_1} &= \left(\Lambda_{h_1}^{h_2} - \Lambda_{h_2}^{h_1}\right) d_{h_1,h_2} \\
&\vdots \\
C_{h_{H-1}} - C_{h_{H-2}} &= \left(\Lambda_{h_{H-2}}^{h_{H-1}} - \Lambda_{h_{H-1}}^{h_{H-2}}\right) d_{h_{H-2},h_{H-1}} \\
C_{h_H} - C_{h_{H-1}} &= \left(\Lambda_{h_{H-1}}^{h_H} - \Lambda_{h_H}^{h_{H-1}}\right) d_{h_{H-1},h_H}
\end{aligned}$$

By summing up the previous equations,

$$C_{h_H} - C_{h_0} = \sum_{i=1}^{H} \left(\Lambda_{h_{i-1}}^{h_i} - \Lambda_{h_i}^{h_{i-1}}\right) d_{h_{i-1},h_i}, \quad (6)$$

were $h_H = v$ and $h_0 = u_{opt}$.

It can be observed from Figure 6 that $|tree_{h_{i-1}}^{h_i}|$ increases as i increases. In particular, for $i_1 < i_2$, $tree_{h_{i_1-1}}^{h_{i_1}} \subset tree_{h_{i_2-1}}^{h_{i_2}}$. On the other hand, $|tree_{h_i}^{h_{i-1}}|$ decreases as i increases ($tree_{h_{i_2}}^{h_{i_2-1}} \subset tree_{h_{i_1}}^{h_{i_1-1}}$). Eventually, $\Lambda_{h_{i-1}}^{h_i}$ increases and $\Lambda_{h_i}^{h_{i-1}}$ decreases as i increases. Consequently, $\Lambda_{h_{i-1}}^{h_i} - \Lambda_{h_i}^{h_{i-1}}$ increases as i increases.

Given the fact that for $i = 1$, $\Lambda_{h_0}^{h_1} - \Lambda_{h_1}^{h_0} = \Lambda_{u_{opt}}^{h_1} - \Lambda_{h_1}^{u_{opt}}$, according to Lemma 1, $\Lambda_{u_{opt}}^{h_1} - \Lambda_{h_1}^{u_{opt}} > 0$. Therefore, $\Lambda_{h_{i-1}}^{h_i} - \Lambda_{h_i}^{h_{i-1}} > 0$, for any i. Consequently, it is concluded that the cost along a path from any node in the network towards the optimal service node, is monotonically decreasing. Given that $\Lambda_{h_{i-1}}^{h_i} - \Lambda_{h_i}^{h_{i-1}} > 0$ and based on Lemma 1, Theorem 2 and the description of the proposed policy, it is evident that the service will always move from node h_i to node h_{i-1}. Eventually, it will arrive at node h_0, which is the optimal service node u_{opt}. ∎

The previous analytical results have shown the efficiency of the proposed service migration policy for a static environment. The fact that under the proposed

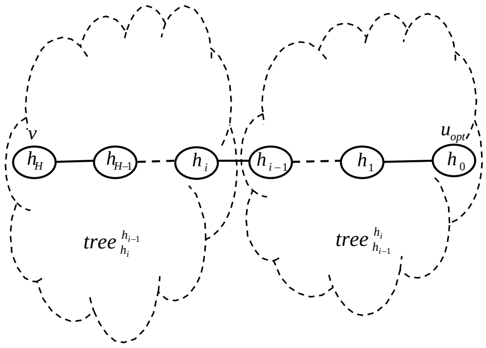


Fig. 6. Examples of $tree_{h_i}^{h_{i-1}}$ and $tree_{h_{i-1}}^{h_i}$.

policy the service is gradually moved in the network allows for certain changes of the network status to be taken into account, thus adapting the direction of the service movement towards a new optimal service node, whenever it is required. Consequently, the proposed policy is suitable for dynamic environments as well.

VI. Some Practical Considerations

In this section some practical issues are discussed and some ideas as to how to cope with them are presented.

The proposed Service Migration Policy - as it was presented so far - assumes that the service moves from node to node until it reaches the optimal service position. However, there might be networks that this is not realistic either due to a certain "cost" to install the service to a new node or due to the fact that some nodes may not be suitable for hosting the service (e.g. limited processing power, physical memory, bandwidth, etc.). Given the aforementioned limitations, may be the only feasible approach could be to move the service only once in order to place it at the optimal service node (assuming that the latter is suitable to host the service).

Under the aforementioned constraints, an alternative approach that can be considered is to move not the actual service but a *service monitoring entity* (SME). The role of this SME is to monitor the service request process at the locations visited and apply the proposed Service Migration Policy (referred to hereafter as the SME Migration Policy) and eventually reach the optimal SME (and service) node location. At the end of this process the SME will notify the original service node of the optimal service node position. At each node, the SME will enable the monitoring process and based on its results, it will determine the next hop. Finally, the service will be placed at the optimal position without being installed in all intermediate nodes (e.g., see Figure 6).

Service movement can also be accelerated by installing a monitoring process at each node in the network, thus, have available the aggregate service demand associated with each node. When the service (or an entity similar to SME previously discussed but with no monitoring functionality) moves to a new node, is already aware about the aggregate service demands of its previous location. For the example depicted in Figure 4, when the service is located at node y, the aggregate service demands Λ_y^z are available. When it moves to node z, the amount of data carried through the link between node y and node z will be the aggregate amount of data for all node in $tree_y^z$ and therefore, the aggregate service demands will be equal to Λ_y^z. Assuming that the estimation regarding Λ_y^z is carried from node y to node z, the direction of the inequality of Lemma 1 can be determined immediately, using the estimations of the monitoring process at node y. Eventually, the service is capable of moving much faster than before to the optimal service node (no need to wait for estimations regarding a piece of information conveyed from the previous node). However, this approach assumes that all nodes in the network continuously monitor the aggregate service demands which may not be realistic for some networks.

The estimation of the aggregate service demands at the service node is important since the service movement depends on their values. Consider the case where node y is the service node and node z is a neighbor node, $z \in S_y$. The aggregate service demands for $tree_z^y$ are given by Equation (3) and denoted by Λ_z^y, $\forall z \in S_y$. It is important to understand how the service node becomes aware of Λ_z^y. One possible approach would be for all nodes u in the network to send messages to the service node regarding their service demands λ_u. However, apart from the fact that this approach requires a large number of messages ($O(N)$), the determination of λ_u by each node u may be fairly inaccurate since there may not be an adequate amount of data associated with a given node and service to estimate accurate mean values λ_u (demands per unit time). Eventually, the estimated aggregate service demands may not be as accurate as it should.

An alternative approach would be to estimate Λ_z^y (i.e., the aggregate service demands corresponding to any set of $tree_z^y$ for any pair of nodes y, z in the network) based on *aggregate statistics* regarding the amount of data forwarded in the network between the service node and each neighbor node that correspond to the particular service. Let $\tilde{\Lambda}_z^y$ denote the aforementioned estimation Since $\tilde{\Lambda}_z^y$ is the sum of a number of random variables generated by typically a large number of nodes in $tree_z^y$, it is evident that its estimation can be more accurate than that of each λ_u, $u \in tree_z^y$, and faster (more aggregate demand events per unit time).

It is important to note that even if the estimate $\tilde{\Lambda}_z^y$ is not very accurate (e.g., because the afforded sampling horizon is small), its inaccuracy is very likely not to impact on the correct operation of the proposed Service Migration Policy since the latter considers differences in the values Λ_z^y associated with neighbor nodes and not actual values.

In view of Equation (6) from Theorem 3, it can be observed that when the number of hops between the service node and the optimal service node is likely to be relatively large (i is large), then the difference $\Lambda_{h_{i-1}}^{h_i} - \Lambda_{h_i}^{h_{i-1}}$ is relatively large as well. Therefore, a relatively small time horizon may be sufficient to derive sufficiently accurate service demand rates and move the service location to the next node. When the number of hops is relatively small (i is small), $\Lambda_{h_{i-1}}^{h_i}$ is likely to be closer to $\Lambda_{h_i}^{h_{i-1}}$. Therefore, a relatively large time horizon may be required to derive sufficiently accurate aggregate demand rates and move the service location to the next node.

VII. Conclusions

The efficient service placement problem for the tree topology case was studied in this paper. This problem, identical to the 1-median problem, is hard to be solved since existing approaches require global information and $O(N^2)$ messages. As a result these approaches are not suitable for dynamic environments.

In order to cope with this inefficiency, the idea of service migration was exploited in this paper. Initially, analytical results have shown that in order to determine the cost difference among neighbor service nodes global information is not required. Information regarding the aggregate service demands of a certain set of nodes can easily become available at the current service node using a simple monitoring process. It was proved that this information is adequate for determining at which neighbor node the service cost will decrease. It should be noted that this result was also proved to be valid for a general network topology.

The aforementioned analytical results and the subsequent observations were the motivation behind a simple migration policy that is proposed here. This policy allows for the service movement among neighbor nodes. It is exclusively based on the aggregate service demands information (always available at the service node) and therefore, it is scalable, of low complexity (no message exchanges) and can easily adapt to any changes of the network status (suitable for dynamic environments).

Regarding a static environment (as it is the case considered by the existing solutions for the 1-median problem), it was analytically proved that if the service is moved under the proposed policy, it eventually arrives at the optimal service location. Moreover, the cost decreases monotonically along the path over which the service is moved towards the optimal service node. Consequently, the solution provided by the proposed simple policy is the optimal one (identical to the solution of the 1-median problem).

Implementation and convergence issues have also been discussed. The estimation of the aggregate service demands was proposed to take place by monitoring the amount of data associated to the particular service.

Finally, as it is stated before, this paper studied the undirected tree topology case, which in itself is rather important in communication networks, where data packets are forwarded towards their destination over the branches of a tree defined by the employed routing protocol. As it was shown here, some of the results may be used for the general topology case, while others can be extended for the latter case. Thus, inspired by the analytical results presented in this paper, further future work will focus on the p-median problem for the general network topology case and try to provide for a scalable, low complexity solution suitable for dynamic environments, such as ad hoc and autonomic networks, exploiting the idea of service migration.

References

[1] 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.

[2] `http:\\www.cascadas-project.org`

[3] `http:\\www.bionets.org`

[4] `http:\\www.ana-project.org`

[5] P.B. Mirchandani and R.L. Francis, "Discrete Location Theory," John Wiley and Sons, 1990.

[6] 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.

[7] A. Tamir, "An $O(pn^2)$ algorithm for p-median and related problems on tree graphs," Operations Research Letters, 19 (1996), 59–64.

[8] 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.

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

[10] 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.

[11] V. Arya, N. Garg, R. Khandekar, K. Munagala, 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.

[12] 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.

[13] 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.

[14] D. Bertsekas and R. Gallager, "Data networks," 2nd edition, Prentice-Hall, Inc., 1992.