Scalable Communication Cost Reduction: The Chord Case

Konstantinos Oikonomou, Spyros Sioutas, and Ioannis Stavrakakis,

***Abstract*— In *peer-to-peer* (P2P) network system design a main focus is on efficient *service discovery* schemes, most frequently assuming permanent (or long-term stationary) positions for *service facilities*, neglecting *communication costs* due to the actual locations of the facilities. Since the problem of communication cost minimization is a large optimization problem (NP-hard) and requires *global information* (i.e., *not scalable*), in this paper the *service migration* philosophy is adopted which permits service facility *movements* and yields smaller communication cost in a *scalable* manner (i.e., based on *local information*). Service migration is incorporated in the *Chord* P2P system, imposing certain changes (e.g., the extension of the service discovery scheme of Chord) and introducing an *extra system overhead* (i.e., *update messages*) for the efficient operation of the (enhanced) system. As it is demonstrated here using simulation results, the communication cost corresponding to the extra system overhead is significantly small and more than enough compensated for by the communication cost reduction due to the introduction of service migration. As long as the network remains unchanged, the former communication cost is *paid only once* while the latter communication cost reduction is *permanent*.**

I. Introduction

Peer-to-Peer (P2P) overlay networks are widely employed and used in modern large-scale and dynamic network environments providing for a new networking paradigm in which the offered service (e.g., content) is distributed among the network peers. Following the pioneering example of Napster in 1999, [1], several P2P systems have been proposed, [2], the most popular among them being Pastry, [3], Tapestry, [4], Chord, [5], Gnutella, [6].

P2P networks are usually categorized as *structured* (e.g., [3], [4], [5]) or *unstructured* (e.g., [6]) depending on the particulars of the implementation, [2]. In unstructured P2P networks, the service offered to the users can be located under no or loose rules, whereas in structured P2P networks the offered service is placed at specific locations in a way that facilitates its subsequent discovery.

A basic difference between unstructured and structured P2P systems amounts to the increased overhead induced by the latter ones, that is compensated for by the efficiency of its *service discovery* process, especially for rare items. In particular, in structured P2P systems distributed hash tables (DHTs) are used for scalable storage and retrieval of information (e.g., the location of a service facility) at the *overlay* (or peer) nodes. Whenever a request for the location of a service is applied, the DHT-based service discovery scheme is capable of identifying the overlay node responsible for the service and thus, retrieve the service facility locations. Due to the structured nature, arrivals, departures or movements of the offered services are costly in general (further details provided in the sequel in Section II) in structured P2P systems.

The particular locations of service facilities are frequently the source of some – often neglected – cost that corresponds to the *communication cost* between the particular *facility node* and the user being served. The derivation of the optimal facility locations in a network (i.e., locations yielding the minimum communication cost) is a problem that requires *global information* and is NP-hard in the general case, [7]. Even though nowadays network nodes are capable enough to host a large variety of service facilities (e.g., ftp servers) – almost unthinkable a decade before – the requirement for global information is prohibitive as well as the recalculation of the large (NP-hard) optimization problem in response to changes of the particular dynamic and large-scale network environment.

Distributed approaches have been proposed lately, (e.g., [9], [10], [11], [12], [13], [14], [15]) trying to avoid the aforementioned large optimization problem based on *local information*, [16]. With the exception of the *service migration* approach in [12], these approaches introduce a certain overhead, [16], apart from the modifications required to efficiently incorporate each of them in an existing P2P system. Service migration is an exception since it allows for a single service facility to move to neighbor nodes and reduce the overall communication cost based on a *pathetic* monitoring mechanism requiring no more information than that already available at the facility node, [16]. However, changes to the P2P system – in which service migration is to be incorporated – are still necessary.

In the work presented here, service migration – as proposed in [12] – is assumed for moving service facilities in a (large-scale and dynamic) structured (DHT-based) P2P network in order to reduce the communication cost. Chord has been chosen as a case study since it is a well-known and widely spread P2P system. Note that the ideas proposed here may be utilized in other structured DHT-based P2P networks.

In order to accomplish the goal of reducing the communication cost in Chord using service migration, the traditional environment (of stationary service facilities) is reconsidered

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 Spyros Sioutas are with the Department of Informatics, Ionian University, Greece. Address: Tsirigori Square 7, 49100 Corfu, Greece. Phone: +30 26610 {87708,87706}, Fax: +30 26610 48491, E-mail: {okon,sioutas}@ionio.gr.

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

allowing facilities to move from node to (neighbor) node. Furthermore, the (DHT-based) service discovery scheme of Chord is enhanced and extended to accomodate migrating service facilities. Despite the almost negligible overhead of service migration, its incorporation into Chord requires some modifications and extensions that introduce some *extra system overhead* (e.g., for keeping updated the Chord specific structures regarding the current position of the service facility) that are investigated and evaluated in this paper through simulations. As it is shown here, this extra system overhead is sufficiently compensated for by the benefit of the overall communication cost reduction. The first cost *paid only once* while the latter benefit is enjoyed for as long as the environment remains roughly unchanged. Another contribution of this paper is the proof that communication cost reduction is achievable for any (connected) network topology for any number of service facilities providing a given service (as opposed to the one service facility case studied in [12]).

The rest of this paper is organized as follows. Background information about Chord and service migration is given in Section II. In Section III the proposed enhanced scheme is presented and a discussion about the introduced overhead is included in Section IV. Simulation results, presented in Section V, reveal the fact that the induced extra system overhead – when service migration is employed – is almost negligible when compared to the communication cost savings. The conclusions are drawn in Section VI. In the Appendix a proof is given on the efficiency of service migration in reducing the communication cost in the case of multiple service facilities in the network.

II. Chord P2P and Service Migration

The basic elements of Chord and service migration – needed later in this paper to describe the enhanced scheme for communication cost reduction – are briefly presented next.

Chord bases its operation on a set of network nodes that being arranged in a virtual *overlay circle* (see Figure 1) have certain capabilities and perform certain operations (lookup, join, departure, etc.), [5]. The number of nodes n in the overlay network is normally of the order $O(\log N)$, where N is the total number of nodes in the network, [5].

A basic component of Chord is the *finger table*. Each overlay node maintains information about (at most) m other neighbors, called *fingers*, in a finger table. The i-th finger node is the first overlay node on the overlay circle that succeeds the current node by at least 2^{i-1} nodes, where $1 \leq i \leq m$. In other words, a finger is equal to the *successor* of the operation $(nodeID + 2^{i-1}) \mod 2^m$, where $1 \leq i \leq m$ ($m = O(\log n)$).

Finger tables are extensively used for the Chord overlay operation and the management of node joins, departures, and failures. For example when a node wants to join the overlay network, it has to know at least one node that is already in the network. The joining node chooses an identifier (e.g., the node's IP address) from the identifier space (0 to $2^m - 1$) and sends a join message with this identifier to the node it knows in the network. The join message is routed across the overlay network until it reaches the node that is the successor of the new node based on its chosen identifier. The joining node is inserted into the overlay network at this point and takes on part of the successor node's load. The new node constructs its finger table and the direct neighbor entries to its successor and predecessor nodes. It also updates the finger tables of other nodes in the system that should point to itself. The cost for a node joining the network is $O(\log^2 n)$ *overlay messages* (an overlay message may correspond to more than one messages since neighbor nodes in the overlay network are not necessarily neighbors in the original network).

Similarly, when a node leaves the system, the finger tables of nodes that have entries pointing to the leaving node have to be updated. The cost for updating these tables is $O(\log^2 n)$ overlay messages (the same as the cost of a join). When a node fails, the finger tables that have entries pointing to the failed node will be incorrect. In order to maintain correct finger tables, each node periodically runs a stabilization algorithm where it chooses a random entry in its finger table and updates it.

The service migration philosophy, as described in [12], is adopted in this paper to propose (in the following section) an enhancement to the Chord P2P system. Let K_t be the set of service facility nodes associated with a given service at time t. For the rest of this paper t will be referring to facility movements under the adopted service migration policy. Assuming a single service facility in the network (located at the facility node) the corresponding data of the nodes' *service demands* are forwarded between the served nodes and the facility node over some shortest path. For a (connected) network topology represented by a graph $G(V, E)$, where V is the set of nodes ($N = |V|$) and E the set of links, let $d(u, v)$ denote the *distance* between node u and node v over a shortest path in the network (i.e., the summation of the individual link *weights* over the particular shortest path). It is assumed that $d(u, v) > 0$, for any $u \neq v$. Let λ_u correspond to the service demands of node u. Assuming the service facility is located at some node x, the mean communication cost incurred for serving node u is given by the product $\lambda_u d(u, x)$, [7]. Eventually, the average communication cost $C_t(x)$ incurred when a single service facility is located at some node x at some time t is given by, [7],

$$C_t(x) = \sum_{\forall u \in V} \lambda_u d(u, x). \quad (1)$$

For the case of more than one service facilities, the overall communication cost C_t at time t is given by $C_t = \sum_{\forall x \in K_t} C_t(x)$. The identification of the particular set of facility nodes for which the overall communication cost is minimized is an NP-hard problem in the general case (for example in trees and for one service facility, the complexity of the problem is $O(N^2)$, [8]) and requires global information.

Instead of solving this large optimization problem that requires global information, service migration assumes that the facility nodes are capable of monitoring the *aggregate* service demands – incoming and outgoing – that are forwarded over each link connecting the particular facility node and its neighbor nodes. Based on the relative values of these aggregate service demands (as shown next) service migration is capable of moving a service facility to neighbor nodes, so as to yield a smaller communication cost. Let S_x denote the set of neighbor nodes of node x. Assuming x to be a facility node, each node $y \in S_x$ forwards (incoming and outgoing) data packets corresponding to aggregate service demands denoted by $\Lambda_t(y)$. Based on these definitions, the *Migration Policy*, proposed in [12] and adopted here, is the following.

The Migration Policy: The service is moved from node x to the neighbor node y at time t, $y \in S_x$, iff $\lambda_x + \sum_{\forall u \in S_x \setminus \{y\}} \Lambda_t(u) < \Lambda_t(y)$.

According to the analysis presented in [12], it is ensured that for a service facility movement under Migration Policy at time t from node x to neighbor node y, $C_{t+1}(y) < C_t(x)$. This is also valid for more than one service facilities as it is shown in the Appendix (Theorem 1).

In view of the Migration Policy two interesting observations are possible regarding the difference in 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. Second, it depends on the *difference of the aggregate service demands*. Consequently, it is evident 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. What is actually required is information regarding the aggregate service demands (e.g., $\Lambda_t(y)$) at the facility node (e.g., node x), facilitated by a suitable monitoring mechanisms, as already mentioned. Furthermore, service facilities are inherently adapting their movements towards directions of further cost reduction after changes in the topology and/or service demands due to the dynamic idiosyncracy of the considered network environment. These interesting properties of the Migration Policy are exploited in the sequel to propose an enhancement to the Chord P2P system.

III. The Proposed Scheme

In this section, the proposed enhanced Chord-based system is presented. As already mentioned, the motivation behind the proposed scheme is to exploit the migration philosophy that yields small overhead in the established Chord environment in order to allow for overall communication cost reduction.

It is assumed that services have *unique IDs* and are mapped to overlay nodes based on their IDs. A service is mapped to the first overlay node whose ID is equal to or follows its key. Let the *corresponding overlay node* be the particular overlay node at which a search query for a certain service ID will eventually end up. The corresponding overlay node maintains a *service facility table* containing entries for the location of the service facilities for the particular service ID. For example, as depicted in Figure 1, if node u hosts a certain service facility of some service ID, then the corresponding overlay node (i.e., node w) maintains a corresponding entry in its service facility table. When a search query is launched, it contains a service ID and the overlay is responsible for locating the particular corresponding overlay node (by consuming $O(\log n)$ overlay messages).

Assume now that there is a single service facility in the network at time $t = 0$ located at some node u (i.e., $K_0 = \{u\}$). This node hosting the service facility may or may not be a node of the overlay (Chord) network. In any case, the corresponding overlay node (i.e., node w), after receiving a query for the particular service ID, consults its service facility table, and replies, accordingly (i.e., points to node u).

Under service migration, the facilities are expected to move from node to node attempting to reduce the overall cost, as already described in the previous section. Assume that a service facility movement took place and at time $t = 1$ the particular service facility is located at node u' (i.e., $K_1 = \{u'\}$). This change of the facility location *should be communicated* to the corresponding overlay node w in order to update its service facility table entry (u' replaces u). For this purpose, a special message is sent by node u to node w as soon as the service facility is decided to move to neighbor node u'.

Such an example is depicted in Figure 1. In Figure 1.a the overlay node w is depicted as part of the overlay network, connected to various other network nodes. Node u hosts the particular service (which the overlay node w is responsible for), depicted by a dotted hexagonal around node u. The service facility table of node w is also shown, highlighting the table entry corresponding to the particular service ID and the corresponding node u. In Figure 1.b, the service facility has moved to node u' (the dotted hexagonal now around node u) and node u has sent a special message to inform the corresponding overlay node w. As it is illustrated, the service facility table becomes updated (entry u is replaced by entry u').

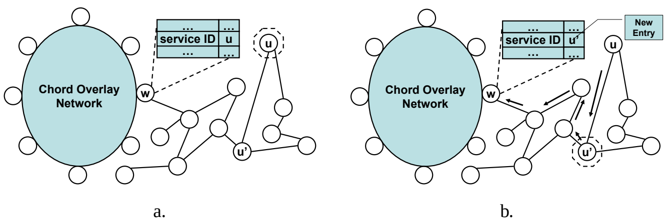


Fig. 1. Example of Chord overlay network and update after service migration. The dense arrows represent messages sent from node u to inform the overlay node w about the service facility movement to node u'. The service facility is depicted by a dotted hexagonal around the node that is located at.

This *transition phase* due to facility movements, may cause problems to nodes (or users) that have applied search queries shortly before the particular movement. These nodes may have already received a reply from the overlay node w that the node offering the particular service is node u and they may try to use this service by contacting node u. Node u will probably drop these (justified) requests for service unless some care is taken to redirect them and inform accordingly the particular nodes about the new location of the service. The same applies for those nodes that were using the facility (e.g., exchanging data) when it was located at node u and continue to use it – preferably uninterrupted – when it moves to node u'.

Finally, note that a single service facility has been considered so far to simplify the presentation of the proposed enhanced Chord scheme. If more than one facilities are employed in the network for a particular service, the previously presented scheme can still be used, apart from some minor changes. For example, the service facility table maintained at the corresponding overlay node contains additional fields regarding the locations of all existing facilities of the particular service in the network.

IV. Requirements and Overhead

The efficient incorporation of service migration in the Chord P2P system – in order to allow for communication cost reduction – requires certain enhancements and introduces in itself some overhead which can be categorized as being due to (a) nodes' capabilities; (b) overlay nodes' capabilities; and (c) the update messages after service facility movements. These three categories are discussed next.

Service migration assumes that all network nodes are *capable of hosting* a facility of a certain service. This is true in most of the cases due to the widespread powerful machines, even considering small devices like mobile phones. On the other hand, some services (e.g., updates of operating systems, antivirus updates) may not be allowed to be hosted by nodes other than those qualified by the service provider. However, in P2P networks the latter is rarely the case and the majority of the provided services fall in the first category.

In addition, service migration requires a monitoring mechanism to provide for estimates of the aggregate traffic for each link of the facility node. Note that light software for capturing data packets is nowadays in common use even in everyday personal computer machines. The fact that estimations about aggregate services demands are needed – and not about each individual node using the service – allows for faster and more efficient estimations, [12].

Note that these requirements regarding nodes' capabilities are not exclusive for the accommodation of service migration in Chord or any DHT-based P2P system, but apply for all cases. More specific to Chord is the overhead due to the extra functionality that should be supported by the overlay nodes, presented next.

As already described in the previous section, in order for Chord to be able to reply to search queries regarding a certain service ID, a new table is introduced (i.e., service facility table) in order for the particular corresponding overlay node to be able to retrieve the location of the service facilities and reply accordingly (or negatively in the case that there is no service facility entry available).

Apart from the memory required for the support of the aforementioned table, the applied search queries – and the subsequent replies – also introduce some overhead due to the exchanged messages ($O(\log n)$ overlay messages). However, this particular overhead is not due to the introduction of service migration (it is due to the support of service lookups inside the Chord overlay network).

Apart from the node capabilities (for both overlay and non-overlay nodes) required to support service migration, the Chord P2P system employs a certain DHT-based mechanism and requires certain *update messages* to be sent in order for this mechanism to remain operational after facility movements. This is clearly illustrated in the example presented in Figure 1 (update messages are sent after a facility movement to inform the corresponding overlay node about the new position of the particular service facility). These (update) messages introduce some *extra system overhead* corresponding to a certain communication cost (depending on the distance between the particular service facility and the corresponding overlay node) that it is compensated (as shown in the simulations section) for by the reduction in the communication cost due to service migration.

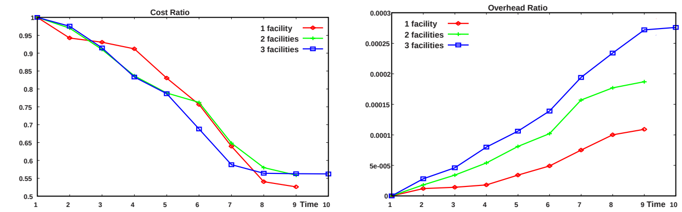
As already mentioned, the transition phase – i.e., before the corresponding overlay node is updated – is a possible source (a) for *denial of service* (e.g., a query is applied when the entry in the service facility table is outdated as it is the case after a service facility movement and before the update message is arrived to the corresponding overlay node); or even (b) for *disruption* of already ongoing service transactions (as it is the case when the service facility moves and some nodes still assume the previous location as the facility node). One possible solution – in both cases – would be to have the previous facility node send update messages regarding the new facility nodes to any node attempting to use the particular service facility. With respect to the first case, the introduced overhead is due to the service requests that have received a reply within the transition phase period. Depending on the case (i.e., the number of queries), this particular overhead may be significant (i.e., large number of queries within the transition phase period) or negligible (i.e., small number of queries). Regarding the second case, the induced overhead increases as the number of nodes (or users) using the particular service facility increases. Clearly, the more popular a certain service facility, the more messages are sent to inform nodes about the new facility location after a facility movement. However, this overhead may be suppressed for this particular case, by *piggybacking* suitable control information in normal data packets still exchanged between the facility and the any other node being served by the facility.

The following section presents simulations results showing the reduction of the communication cost due to service migration and its comparison with the enhanced additional system overhead. Since the study of a service's "popularity" – and consequently, the distribution of applied queries within the transition phase – is an issue with insignificant impact compared to the simulation scenarios presented next, the focus is on the overhead introduced by the update messages in comparison to the communication cost savings due to service migration.

V. Simulation Results

A simulation environment in programming language C is developed for creating network topologies of 10000 nodes (specifically, trees and grids) and implementing the Migration Policy within the Chord P2P system. Each node is randomly assigned a value with respect to service demands. The goal of the simulation results is twofold: to illustrate the overall communication cost reduction and to illustrate the communication cost due to the extra system overhead. For this reason, the results presented here are not averaged values but results of individual simulation experiments. Averaging would have given a macroscopic view failing to give microscopic details about the idiosyncrasies of the migration policies.

The initial position for the facilities is randomly chosen as well as their corresponding overlay node. One, two or three facilities were considered depending on the presented case. All figures on the left (i.e., Figure 2.a and Figure 3.a) present results with respect to the *cost ratio* (i.e., the fraction of the current communication cost over the initial communication cost before the services moved) (y-axis), as a function of time t (x-axis). All figures on the right (i.e., Figure 2.b and Figure 3.b) present results with respect to the *overhead ratio* (i.e., the fraction of the cost due to the extra system overhead over the initial communication cost before the services moved) (y-axis), as a function of time t (x-axis). Time t corresponds to facility movements and starts at time $t = 1$ for each case.



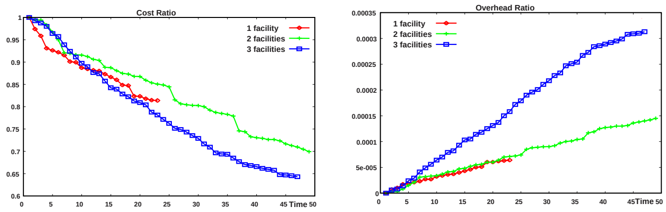
a. b.

Fig. 2. Tree Topology of 10000 nodes.

As it is depicted in Figure 2.a, the overall communication cost monotonically decreases after each facility movement. This is in accordance with the results in [12] for the case of one facility and in accordance with Theorem 1 (see Appendix) for the case of more than one service facilities. Note that the cost ratio drops below 60% of its initial value. This corresponds to more than 40% ($100\% - 60\% = 40\%$) cost savings is *permanent* in the network as long as the topology and/or service demands do not change (i.e., static environment). If they change, then the service facilities will adapt their movement towards more effective positions, [12].

Figure 2.b depicts the correponding overhead ratio for the service facility movements presented in Figure 2.a. As it is obvious, the overhead ratio increases with the number of service movement (as expected, the larger the number of movements, the more the messages need to be sent to update the service facility table of the corresponded overlay node). This extra system overhead is small (i.e., less than 0.03%) when compared to the initial cost (before the service facility started to move) and it is *paid once* (assuming the previously mentioned static environment), while the benefit of the cost reduction (as mentioned before) is permanent.

Tree topologies is a good example with respect to service migration, since there are no cycles to prohibit facility movements to more effective positions, [12]. For example, in tree topologies and for a single service facility in the network, it is ensured that the facility will eventually arrive at the *optimal*



a. b.

Fig. 3. Grid Topology of 100×100 nodes.

position (yielding for communication cost minimization) under the Migration Policy. On the other hand, topologies with cycles in some cases may not allow for a service facility to move even if cost reduction would have been achieved, thus the overall communication cost reduction may not be as much as it may be the case in tree topologies. In order to further illustrate the effectiveness of the proposed scheme in this work, grid topologies are also considered, as it is the case depicted in Figure 3.a. For the case of one service facility, cost savings of about 15% is achieved and the introduced overhead (depicted in Figure 3.b) is about 0.005%. For the case of two service facilities, the extra system overhead is increased (about 0.015%) due to the increased facility movements, but the communication cost reduction is higher (about 30%). Even for the case of three facilities – where the overhead ratio is increased (about 0.035%) – the corresponding savings of the overall cost (i.e., about 35%) compensates more than enough.

Eventually, the overhead paid for updating the service facility tables of the corresponding overlay node is significantly smaller than the reduction of the communication cost. Assuming a static environment, this overhead is paid only once, while the benefits of the overall communication cost reduction are permanent in the system.

VI. Conclusions

In this paper the Chord P2P system and service migration were both considered in an enhanced new DHT-based P2P scheme that permits service facility movements under the service migration philosophy in an attempt to reduce communication costs. The proposed scheme is not applicable only specifically to Chord but can be applied for any DHT-based system; future work aims at other P2P systems, like Pastry, Tapestry, etc. Several applications may be defined, especially in the area of mobile ad hoc networks when service discovery increasingly becomes a issue.

The efficient introduction of service migration requires certain changes of the Chord P2P system (e.g., service facility tables, update messages) in order for the (enhanced) P2P system to remain functional. In particular, when a service facility moves in the network, update messages need to be sent to inform the corresponding overlay node about the new facility node. These update messages introduce some extra system overhead corresponding to a certain communication cost. As it was shown using simulation results, the communication cost corresponding to the extra system overhead is more than compensated for by the communication cost reduction achieved by the employment of service migration (e.g., 0.003% against

35%). Note that the former communication cost is paid only once while the second is permanent as lond as the network remains unchanged. Finally, another contribution of this paper is the proof (included in the Appendix) that service migration allows for communication cost reduction even for the case of more than one service facilities. The presented simulations are in accordance with the analytical findings.

REFERENCES

[1] Napster. [Online]. Available: http://www.napster.com/

[2] Lua, E.K., Crowcroft, J., Pias, M., Sharma, R., Lim, S.: A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. In: IEEE Communications Survey and Tutorial, March (2004).

[3] Rowstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Proceedings of the Middleware, (2001).

[4] Zhao, B.Y., Huang, L., Stribling, J., Rhea, S.C., Joseph, A.D., Kubiatowicz, J.D.: Tapestry: A resilient global-scale overlay for service deployment. In: IEEE Journal on Selected Areas in Communications, vol. 22, no. 1, pp. 4153, January 2004, (2004).

[5] Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup protocol for internet applications. In : IEEE/ACM Transactions on Networking, vol. 11, no. 1, pp. 17 32 (2003).

[6] Gnutella. Gnutella rfc. http://rfc-gnutella.sourceforge.net/, 2002.

[7] Mirchandani, P.B., Francis, R.L.: Discrete Location Theory. In: John Wiley and Sons, 1990.

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

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

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

[11] Moscibroda, T., Wattenhofer, R.: Facility Location: Distributed Approximation. In: 24th ACM Symp. on the Principles of Distributed Computing, Las Vegas, USA, July (2005).

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

[13] Krivitski, D., Schiuster, A., Wolff, R.: A Local Facility Location Algorithm for Large-Scale Distributed Systems. In: Journal of Grid Computing, Volume 5, Number 4 / December, 2007, pp. 361–378, (2007).

[14] Laoutaris, N., Smaragdakis, G., Oikonomou, K., Stavrakakis, I., Bestavros, A.: Distributed Placement of Service Facilities in Large-Scale Networks. In: IEEE INFOCOM'07, May 6-12, 2007, Anchorage, Alaska (2007).

[15] Frank, C., K. Römer, K.: 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).

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

APPENDIX

Theorem 1: In a network of more than one service facilities, if a facility – located at node x at time t – under Migration Policy moves to some neighbor node y, then $C_{t+1} < C_t$.

Proof: Suppose there are more than one facilities in the network (i.e., $|K_t| = c$, for some constant integer value $c > 1$ for any t) employing the Migration Policy. Assume also that at least one facility movement has taken place (i.e., $K_{t+1} \neq K_t$) between time t and time $t + 1$. Let $f_t(v)$ denote the facility node corresponding to any network node v at time t. Let set Z denote a set of nodes that at time $t + 1$ are served by a facility different than the one they were served at instance t, or $Z = \{z : \forall z \in V \text{ and } f_{t+1}(v) \neq f_t(v)\}$. The sequel of this proof is based on the fact that some nodes choose to be served by different facilities – after a facility movement – because of smaller distance than before, thus yielding for smaller contribution to the overall cost.

The case that no node changed its corresponding facility or $Z = \emptyset$, is trivial, and the results of the analysis presented in [12] can be reused. Assume, now, the case that there some changes of facilities nodes, or $Z \neq \emptyset$. Let C'_{t+1} represent a *hypothetical cost* at time $t + 1$ assuming all nodes $v \in Z$ forced to continue to be served by the same facilities of time t (i.e., K_t instead of K_{t+1}). Given that a service movement has already taken place under Migration Policy, $C_t > C'_{t+1}$ is satisfied. However, $C_{t+1} = C'_{t+1} + \Delta$, 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 $\Delta = \sum_{\forall v \in Z} \lambda_v d(v, f_{t+1}(v)) - \sum_{\forall v \in Z} \lambda_v d(v, f_t(v))$, or $\Delta = \sum_{\forall v \in Z} \lambda_v \big(d(v, f_{t+1}(v)) - d(v, f_t(v))\big)$. However, $d(v, f_t(v)) > d(v, f_{t+1}(v))$, which is the actual reason for nodes $v \in Z$ to change facilities. Therefore, $\Delta < 0$ and consequently, $C_{t+1} < C'_{t+1}$. Eventually, $C_{t+1} < C_t$, and the theorem follows. ∎