

Self-Organized Disjoint Service Placement in Future Mobile Communication Networks

Shah Nawaz Khan^{1,2}, Ali Diab¹, Christian Brosch¹, Mushtaq Ahmad²

¹Ilmenau University of Technology, Gustav Kirchoff Str. 98693, Germany

²GIK Institute of Engineering Sciences & Technology, Topi Swabi, KPK, Pakistan
(shahnawaz, mushtaq)@giki.edu.pk, (ali-diab, christian-brosch)@tu-ilmenau.de

Abstract- Future mobile communication networks will offer many ubiquitous services to its clients such as voice and video communication, access to data and files, use of virtual resources in cloud, etc. The provision of these services in face of the different challenges posed by future wireless networks such as changing network topology, variable load conditions, clients' distribution, QoS requirements etc. is a very difficult task and requires a high degree of self-organization in network operations. One important problem in this context is the self-organized service placement which refers to the problem of finding optimal nodes in the network that are most suitable for hosting a particular service type. An optimal placement of a service and its instances (replicas) not only minimizes the service costs but also reduces the overall network traffic and improves connectivity between clients and servers. This paper proposes a novel network service called Self-Organized Disjoint Service Placement (*SO-DSP*) service which manages other network services and their instances in order to achieve overall network optimization while keeping the individual service's quality at the same level for its clients. The clients of *SO-DSP* are not the end-users of the network but the offered network services.

Keywords: Service placement, Wireless networks, Self-organization, Network management,

1. Introduction

Changing network topology, varying load distribution, different services, and unpredictable client distribution are some of the well-known dynamics of future mobile communication networks. These networks must provide ubiquitous access to information and services in face of many challenges regardless of time and location. In order to achieve these ambitious goals, a high degree of self-organization is required in the network to not only cope with the different network dynamics but also maintain an acceptable level of Quality of Service (QoS). Figure 1 presents an abstract skeleton of future networks scenario where different types of wired and wireless networks are connected to the core framework of Internet. An important feature at the user-end of the future networks scenarios is heterogeneity and mobility. Although the core framework of internet is expected to be optimized for support of different types of services and quality assurance, it will retain its hierarchical structure. One of the most important problems in the context of future mobile communication networks is the problem of self-organized service placement within the network. Since a service and its instances, also known as replicas, can be placed in many locations inside the network, the service placement problem arises in finding out which location/node is most suitable for the actual placement. As a general rule, the closer the service instance is to its clients, the better is the provision of the quality of service and lower overhead on the network [17]. This problem is further aggravated by the fact that a chosen location for service

placement may not remain optimal for a longer period of time due to the different network dynamics [18].

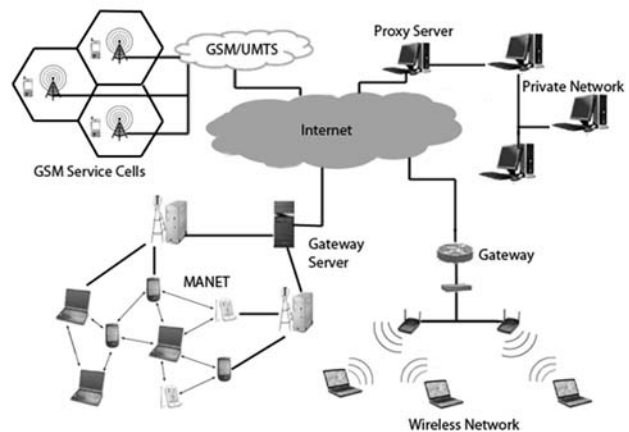


Figure 1. An abstract future network skeleton

Moreover, a network provides a number of services and hence the service placement problem is not limited to a single service but includes the management of all services offered by the network. An optimal placement of a service and its instances not only reduces the service cost but also reduces the overall network traffic thereby, improving the connectivity between clients and servers. The different services offered by a network to its clients are commonly referred to as Disjoint Services. Disjoint Services are those which do not communicate with each other for its service provision. Each of the offered service usually has its own service placement mechanism through which it manages the service and its instances. The network services are however, not entirely independent as they reside and utilize the resources of the same core framework of Internet and hence the actions performed by one service for its placement may affect other services in an adverse way. It may further force other services to perform actions on their part in order to adapt to the new situation. This highlights the fact that there is a real need of a network service for the management of other disjoint network services. Furthermore, the proposed management service has to work in a self-organized and fully distributed manner in order to fulfill the requirements of future dynamic mobile communication networks.

In this paper we propose a novel service called Self-Organized Disjoint Service Placement (*SO-DSP*) which is a self-organized and distributed service for the management of other network services and their instances with the main goal of achieving better network resource utilization while keeping the quality of service of all services at an acceptable level for its clients. *SO-DSP* performs distributed management of other network services and aims at achieving

an optimal placement and replication/migration of these services.

The rest of the paper is organized as follows. Section II gives a detailed insight into the related work followed by detailed description of our proposed *SO-DSP* service in section III. We present our simulation analysis and preliminary results in section IV. The paper concludes in section V concludes with a brief summary and future work.

2. Related Work

The service placement problem is not entirely new and has been addressed in the context of traditional internet-based service provision. Many service placement schemes have been proposed which can broadly be classified into three categories i.e. centralized, distributed, and hybrid service placement techniques. The centralized techniques use a central entity to control the service placement and related management tasks and usually requires prior knowledge about network topology, service holders, clients' distribution, etc. The distributed service placement techniques most of the times do not require network knowledge a-priori and take decisions based on local information only which makes them more scalable and robust to attacks and failures. The hybrid approaches of service placement try to combine the advantages of the centralized and distributed techniques. There are however, a limited number of hybrid techniques proposed and much of the proposed work is still theoretical. We present some of the well-known techniques of each category.

2.1 Centralized Service Placement

In [12], the problem of placement of web proxies in the internet is considered. The authors investigate to find the optimal placement of multiple web proxies (M) among some potential sites (N) under a given traffic pattern. The most obvious solution to this problem may be to place proxies on routers for LANs, ISP gateways, or some strategic locations. The authors in [12] model this problem as a dynamic programming problem and find an optimal solution for a tree topology in $O(N_3M_2)$ time. Other techniques for web replica placement are summarized in [16]. The algorithms discussed include Random algorithm, Greedy algorithm, and Client Clustering. The random algorithm places the replicas randomly at different locations in the network and then tries to optimize them according to the clients' requirements. According to [8] the ratio of expected maximum client-replica distance between optimal and random placement increases logarithmically. The greedy algorithm places replicas on trees in a greedy manner. The algorithm has a complexity of $O(nk)$ where k is the number of replicas to be placed and n is the number of nodes in the tree. The client clustering technique places replicas at the higher concentration of clients based on prior knowledge. This algorithm however, as proved in [9] creates very inefficient cluster representations. In [11], the Service Grid architecture is proposed which consists of three core components: Replication Manager (*RM*), Group Manager (*GM*), and Site Manager (*SM*). *RM* performs the basic replica management; *GM* manages the clients while *SM* serves to interact with *GM* to determine network performance between replicas and client groups. The centralized approaches are simple to employ and manage but are limited in their scalability and suffer from a single point of failure. The disadvantages of

centralized techniques such as prior network knowledge requirement, limited scalability, and non-robustness to failures are very severe and hence make them rigid in adaptations for future mobile communication networks scenarios.

2.2 Distributed Service Placement

Replication in Dense MANETs (REDMAN) [2], proposes "replication manager" which is a controlling entity placed in the network by considering local topology. A node hosts the replication manager if it is part of a dense location in the network. The replication manager itself uses a simple heuristic to place read-only resources. In [13], the authors consider a more mobile scenario. The nodes' connectivity to the network is predicted based on their movement parameters. If the nodes' movement will result in network partitioning and disconnection from services, a service replica is created on the nodes moving away from the service range. The replication is aimed at a node that could later serve the newly created partition. Sailhan et al. [15], propose to reduce the service discovery cost in the network by handling service discovery requests locally at a nearby directory. A node with no access to a directory broadcasts queries. If no node in the local area hosts a directory, the initiating node selects a node for hosting the directory the selection of which is based on its connectivity and coverage. The authors in [10] consider a few-hop nodes' neighborhood as a small network where topology and service demands are known. They develop a distributed algorithm for solving the un-capacitated k -median and the facility location problem. A mapping is done to compensate for nodes outside the hop count threshold. They use the centralized algorithm in a distributed manner by limiting the network topology with some service overlapping at the edges of the limited topology. The un-capacitated facility location problem is also considered in [4]. They propose a distributed algorithm that iteratively negotiates the service assignment among nodes which may host services (based on prior knowledge). Each of these nodes evaluates its local topology and calculates service provisioning cost which is then propagated to the clients. Clients select the service with the least cost. The same problem is considered by the authors in [14]. They present a distributed algorithm and study the trade-off between communication overhead and quality of approximation with the assumption that potential service hosts can communicate with all clients. The authors describe facility location as a fractional linear program which can be approximated in a definite time. Distributed randomized rounding is used to map the results of the first iteration to the actual un-capacitated facility location problem. Distributed service placement strategies offer more promising solutions to the service placement problem. In [19] the authors present a service placement technique for ad hoc networks called SPi. The proposed algorithm tries to find the best places for services placement taking into consideration the multi-hop scenario of ad hoc networks. Most of these techniques are scalable and do not require global network knowledge. The self-organized nature of these algorithms is highly suited to the needs of future communication networks.

2.3 Hybrid Service Placement

The cluster heads as service provisioning entities are considered in [5]. The authors apply the concept of the Low Energy Adaptive Clustering Hierarchy (LEACH-C) protocol [6] and suggest a centralized selection algorithm running on

the base station for facility location. They formulate the problem as an un-capacitated facility location problem which allows for a variable number of cluster heads even on the energy constrained candidate nodes. In [3], the authors propose an algorithm called COCOA which is based on precise network state information. COCOA achieves the performance of centralized algorithms and maintains to a certain degree, the scalability of distributed algorithms. Hybrid service placement techniques provide an intermediate solution between the inflexible centralized approaches and the presently more theoretical self-organized distributed techniques. Table-1 summarizes some of the main features of the three types of service placement mechanisms discussed in this section.

Table 1. Comparison of service placement methodologies

	Centralized	Distributed	Hybrid
Scalability	Low	High	Intermediate
Robustness	Low	High	Intermediate
Prior knowledge	Global	Local	Technique-specific
Vulnerability	High	Low	High
Self-organization	Low	High	Low
Future use	Low	High	Intermediate

3. Self-Organized Disjoint Service Placement (SO-DSP)

Before going into the details of the proposed SO-DSP service and its functionality, it is important to define some of the most common terms related to service placement problem.

- *Service*: A piece of network functionality enabled by software and hardware resources catering for certain requirements of users' needs. Each service has an associated cost in terms of computation, communication, etc.
- *Service migration*: The removal of a service from one physical location and its placement in another location inside the networks based on certain user or organizational requirements. Usually performed for optimizing the service in terms of quality or cost reduction.
- *Session migration*: The process of switching the current clients' sessions associated with a service type to the new location of the service in the event of service migration. The problem of how to migrate the sessions without disruption is a separate problem and out of the scope of this paper.
- *Service replication*: The creation and placement of an exact copy of a service in a different location in the network.

As previously mentioned, to enable different services to coexist in a network and share resources, there is a genuine need for a management service to reside in the network and facilitate equitable resource utilization. This requirement leads us to the development of Self-Organized Disjoint Service Placement (SO-DSP) service which is described in details in this section. We do not assume any global knowledge about the network topology for SO-DSP. The management service needs to work in a fully distributed

manner in order to cope with varying network conditions. The following list defines an abstract network and presents the assumptions made for the proposed SO-DSP service. Most of the assumptions are similar to those in [1].

- The network topology is represented as a graph $G = (V, E)$, where V represents the nodes in the network and E represents the edges between them. An edge between two nodes i and j is represented by $e(i, j)$.
- The network consists of heterogeneous nodes with different hardware and software capabilities i.e. different resources are associated with each node. A particular service type S may not be operable on all network nodes. The nodes capable of hosting service type S are represented as V_S where $V_S \subseteq V$.
- An overlay network of service type S is described by the graph $G_S = (V_S, E_S)$ where E_S is the set of edges between the nodes of V_S . An edge between two nodes $n(i)$ and $n(j)$ is represented by $e_{S(i, j, m)}$ where m denotes the number of physical links between these two nodes. Note that there may exist more than one path between $n(i)$ and $n(j)$.
- The assumed network offers a specific set of service types $SC = \{S_1, S_2, \dots, S_d\}$ where d is the index of the service types offered by the network. Each service type $S_i \in SC$ serves a set of clients H . For simplicity, it is assumed that the clients requesting a service type are located on the nodes of V . This implies that clients' mobility is modeled by changing their distribution among nodes.
- RS_i represents the set of instances of service type S_i . Creation of new or removal of existing instances of a service type is determined by the service placement technique used by the service type. This decision is affected, however, by the decision of the management service (in our case SO-DSP). RS_i is a function of time and $RS_i(t)$ denotes the set of replicas of service type S_i existing in the network at time t .
- When a client requests a service, it is assigned a replica capable of providing the requested service at the minimum cost and required QoS. Adequate replica selection and assignment to the requesting client is a task of a look up service and is out of the scope of the proposed management service since the aim is to manage the resource sharing and coexistence of different types of services.

3.1 Basic Idea of SO-DSP

Like other network services, SO-DSP is a distributed network service such that its service cost is kept at minimum and all its clients are served. As opposed to other services (e.g. A/V streaming, web searches etc.), the clients of SO-DSP are not the users or mobile nodes. Instead, the clients of SO-DSP are other network services offered by the network i.e. $r_j^{S_i}$, $S_i \in SC$. Consider Fig. 2 which represents an example network and a corresponding overlay network providing SO-DSP service. Fig. 2 also shows that the network offers two other services S_1 and S_2 and the nodes capable of hosting the replicas of these service types, V_{S_1} and V_{S_2} . We shall now take a case study based on the example network of Fig. 2 and highlight in Fig. 3 the service replication/migration issue. Fig. 3 shows that there is one replica/instance of SO-DSP (r_j^{SO}

DSP), one replica of service type S_2 ($r_1^{S_2}$), and two replicas of service type S_1 ($r_1^{S_1}$ and $r_2^{S_1}$). There are 20 clients requesting service from $r_1^{S_1}$ via node $n_{(12)}$ which is also serving 15 clients of $r_1^{S_2}$. The requests for service type S_1 are depicted with green arrows while the requests for S_2 are depicted with red arrow. Fig. 3 shows that migration of $r_1^{S_1}$ and $r_1^{S_2}$ will decrease the cost of service and communication for all clients. Let us assume that node $n_{(12)}$ can host only one of these two replicas at a time. Here, the problem arises from the fact that each service type decides independently to do migration/replication which results in many problems. To prevent this with *SO-DSP*, when a replica takes a decision to migrate/replicate to a node $n_{(k)}$, it contacts *SO-DSP* instance which provides it with information about replicas of other service types aiming to do the same migration/replication to $n_{(k)}$, the resources required for these replicas and service hosting capabilities of $n_{(k)}$. This procedure therefore, ensures the coexistence of different service types and sharing of resources.

3.2 SO-DSP: Operation overview

To be able to manage resources among services, each replica of any service type that the network offers has to be assigned to an *SO-DSP* instance (r_i^{SO-DSP}). Let us assume that the set of clients u_j is $\rho = \{u_1, u_2, \dots, u_l\}$, where l is the number of clients of r_i^{SO-DSP} and u_l is a replica of a service type the network offers, where $j = 1, 2, \dots, l$. Each client service u_j provides r_i^{SO-DSP} with information about the service u_j , the priority of this service $P(u_j)$, and the resources u_j required when replicating/migrating $\square(u_j)$. Following that, each replica u_j calculates periodically, based on its current status, the

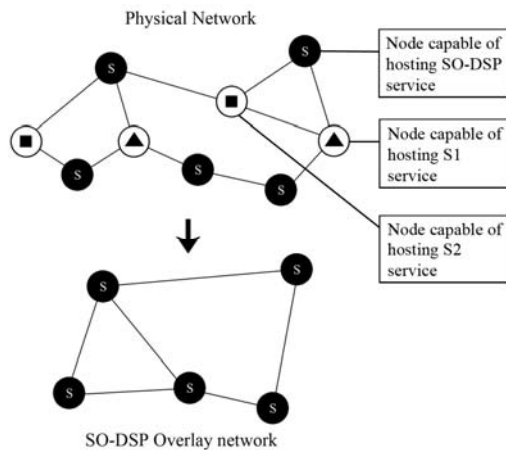


Figure 2. Physical and overlay networks for SO-DSP

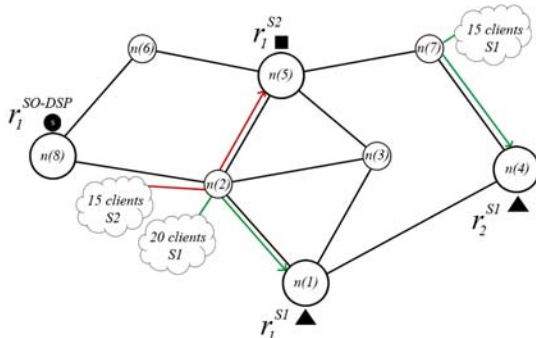


Figure 3. SO-DSP case study

probability to do replication/migration in time frame Δt . This probability is referred to as $P_{rep}(u_j)$. Each replica u_j defines the value of Δt by itself which is a time dependent variable value. The best way is to define this value as a function of the network load. Increasing network load reduces the value of Δt and vice-versa. This guarantees that the replica u_j reacts quickly on network changes while considering the management of its decisions by r_i^{SO-DSP} . When the probability of replication/migration for replica u_j gets high, it starts to contact r_i^{SO-DSP} . For this purpose, periodic messages (each Δt) containing $P_{rep}(u_j)$ and the node $n_{(k)}$, to which the replica intends to migrate, are sent from u_j to the responsible r_i^{SO-DSP} . In this way, the instance r_i^{SO-DSP} has knowledge of the replicas it manages and to which nodes these replicas plan to migrate/replicate. When $P_{rep}(u_j)$ gets too high, meaning that replication/migration should be done in short time, r_i^{SO-DSP} notifies u_j based on $P_{rep}(u_j)$, $P(u_j)$, and $\square(u_j)$, if it can proceed with the replication/migration to node $n_{(k)}$ or if it has to take another decision.

3.3 SO-DSP: placement of instances

Proper placement of *SO-DSP* instances is important so that all service types can have access to a nearby *SO-DSP* instance. For this purpose, we assume that each *SO-DSP* instance has a replication radius of (γ) which limits the number of hops between the instances of *SO-DSP*. Each instance sets the value of γ to a default value when the instance is created. During the lifetime of the instance, the value of γ is updated dynamically based on the number of

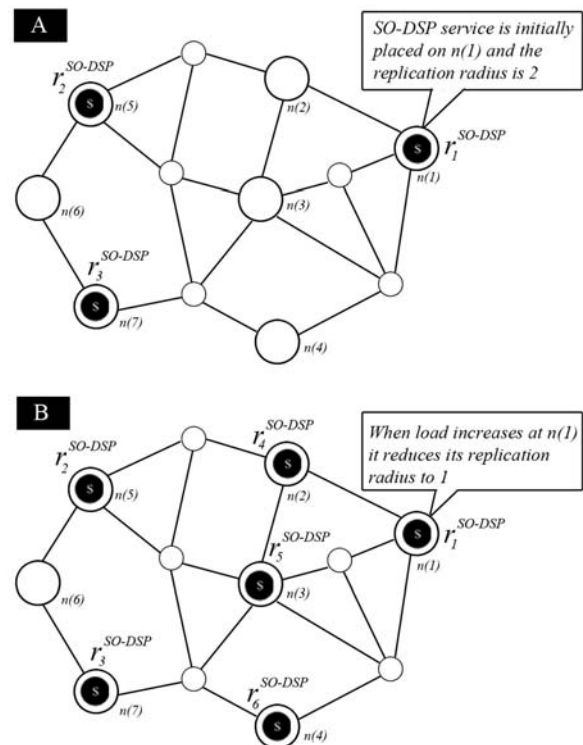


Figure 4. SO-DSP placement of instances

replicas the network contains. Consider the network of Fig. 4 and assume the initial *SO-DSP* instance (r_1^{SO-DSP}) is placed on node $n_{(1)}$. In Fig. 4, the larger circular nodes depict those nodes which are capable of hosting an instance of *SO-DSP* service while the smaller nodes are incapable to do so.

Assume that the replication radius γ for (r_i^{SO-DSP}) is set to 2. It is worth pointing out here that the replication radius is implemented on the overlay network of a service and not according to the physical topology of the network. The first instance i.e. r_1^{SO-DSP} broadcasts *Sync* messages with time-to-live set equal to the value of γ . The purpose of *Sync* messages is to find out if there are other instances of *SO-DSP* in the γ -hop vicinity of $n_{(1)}$ and to synchronize with them if they exist. If (r_i^{SO-DSP}) does not receive a response after g retransmissions, it assumes that there are no other instances of *SO-DSP* in its γ -hop vicinity of itself. As a result, it creates replicas and places them on $n_{(j)} \in V_{SO-DSP}$, where the distance between $n_{(1)}$ and $n_{(j)}$ is γ -hops. In the network shown in Fig. 4, two new replicas (r_2^{SO-DSP}) and (r_3^{SO-DSP}) will be created and placed on $n_{(5)}$, and $n_{(7)}$ respectively. The two newly created replicas will do the same operation as (r_1^{SO-DSP}) to find and synchronize with other existing instances. Each instance of *SO-DSP* dynamically adjusts the value of γ by itself depending on the number of services it manages. This implies that different *SO-DSP* instances may have different values of γ . This makes sense since the *SO-DSP* instances have to cope with the dynamics of the network and that some network nodes may be more loaded than others. If an instance with a replication radius of two finds another instance in its one-hop vicinity, it assumes an over-load condition at a network segment and synchronizes with it. Consider Fig. 4 again and assume that the number of service replicas r_i^{SO-DSP} manages increases. This implies that more instances of *SO-DSP* need to be created. To do this, r_i^{SO-DSP} can simply reduce the value of γ to 1. This will result in replication of more instances of *SO-DSP* in the network and placement of these instances in 1-hop vicinity r_i^{SO-DSP} .

3.4 Summary

The *SO-DSP* is a distributed and self-organized service to manage the coexistence of different services in mobile communication networks. On one side, *SO-DSP* is similar to other network services from the placement point of view since *SO-DSP* should be dynamically placed to minimize overall service provisioning costs. On the other side, *SO-DSP* differs from other known services in two aspects: First aspect is the clients themselves, which are not users or mobile devices but instead, replicas of services which the network offers to the clients. The second aspect is the function which *SO-DSP* offers which focuses on the management of resources among other services. This may prohibit that each of these services can minimize their cost of serving clients. However, the cost of operating all services in the network is minimized. The basic idea of *SO-DSP* is that each replica of any service type in the network is assigned to an *SO-DSP* instance. This replica notifies the *SO-DSP* instance about the resources it needs when replicating/migrating, the priority of the service the replica offers, etc. As the probability of replication/migration of a service type increases, it starts to contact the *SO-DSP* instance to organize the replication/migration with other replicas located around. *SO-DSP* enables minimizing the cost of operating all services in the network as a whole, retaining the Quality of Service (*QoS*) and balancing the load. It works in a distributed manner and does not require global knowledge of the network. The most important features of *SO-DSP* are its capabilities of self-organization, which enables *SO-DSP* instances to cope with dynamic networks where network topology, load patterns; client distribution etc. changes over time.

4. Simulations & Results

In order to validate the proposed *SO-DSP* management service, we performed some preliminary simulations in OMNeT++ simulation platform together with MiXiM model. OMNeT++ is a discrete event simulation system well suited to the evaluation of wired and wireless networks. The MiXiM model provides some physical layer details such as signal propagation characteristics and medium access protocols which are necessary to model for a mobile network scenario particularly for the clients asking for network services. The simulations were performed on a network scenario where 80 heterogeneous nodes constituted the core service provider network and 50 wireless access clients producing the service requests over time. To differentiate between service instances and their hosting requirements, five different service types were simulated in the network with different hosting requirements. The heterogeneity of the service nodes was hard coded into the system via a capability list which determined the nodes' capacity to host a particular type of service instance. The clients were only associated with the

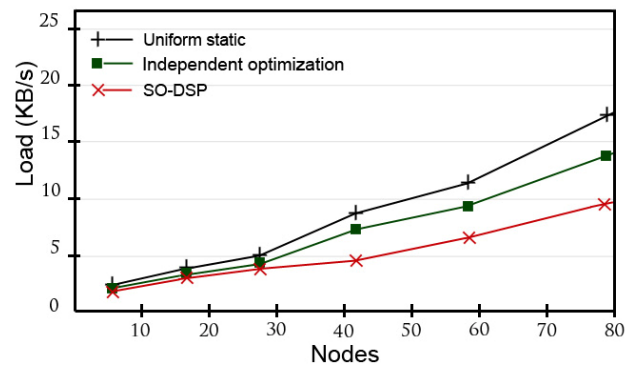


Figure 5. Network load associated with different placement strategies.

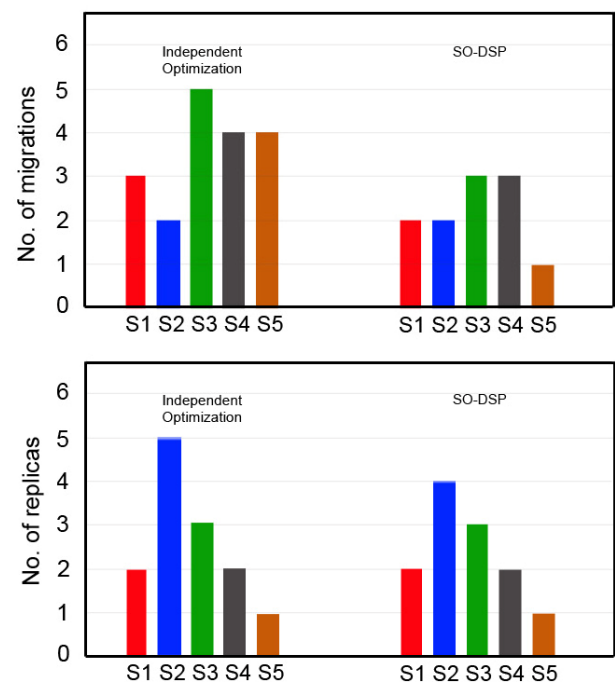


Figure 6. Number of migration and replicas for all service types

end nodes in the network and their distribution was uniform random over a simulation instance. We first analyzed the effect of multiple services operating independently in the network in terms of the network load and compared it with the positive effects of SO-DSP management.

Figure 5 shows the results of three different service management strategies i.e. uniform static; where the service deployment occurs at the beginning of simulation and does not adapt to changing client requests, independent optimization; where service instances of different types compete with each other to optimize their own quality of service using greedy algorithm, and SO-DSP; where the service instance and their associated functions are managed by the SO-DSP management service. As evident from figure 5, SO-DSP reduces the network load considerably in comparison to the other two approaches in our basic preliminary analysis. We can attribute this effect to the knowledge which SO-DSP acquires from all service instances operating in the network and the client distribution. SO-DSP authorizes the service migration or replication for the most suitable

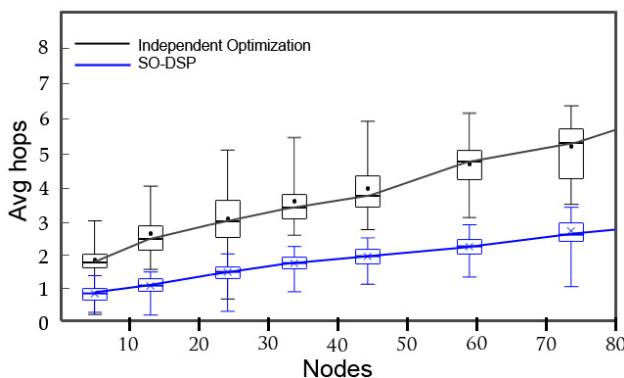


Figure 7. Average distance travelled by service requests to service instances

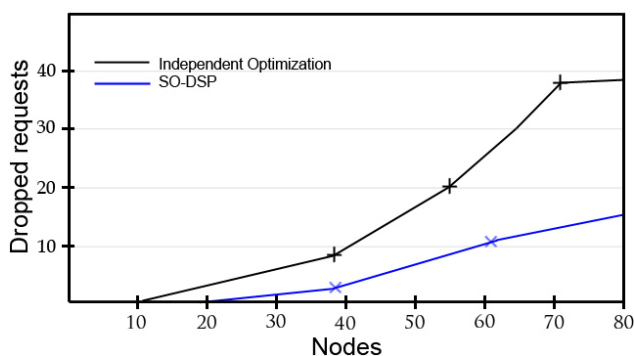


Figure 8. Dropped service requests during simulation for all clients

service type in order to keep the network resource utilization at a higher level.

Figure 6 shows the impact of SO-DSP management on service migration and replication functions against independent optimization performed by all five service types. In the case of service migration, we kept replication radius r to maximum value in order to mitigate the effect of replica creation and evaluate only the migration aspect of service management. Additionally, the distribution of clients was altered during simulation to model mobility of clients which mainly triggered a service migration. SO-DSP reduced the total number of migrations by 39% in the simulated scenario while maintaining the observed QoS at the clients to an

acceptable level. Reducing the number of service migrations has a direct impact on the client's observed QoS and on the overall service costs. Although a service migration is intended to decrease the costs of a service type, it does produce a negative impact on other service types when their requirements are not considered. In the second case of replication, we see a rather minimal impact of SO-DSP management on the number of replicas created during the observed simulation time. This is understandable since replicas are created depending upon the clients' service demands as well as on the capacity of the network to host service instances. The heterogeneity of nodes in terms of their ability to host certain services also contributes to this result. The existence of SO-DSP as an additional network service has minimal overhead in terms of costs. This is due to the nature of SO-DSP service where the clients are not the end users of the network but the offered network services. The management of network services through SO-DSP is only computationally expensive but does not incur costs

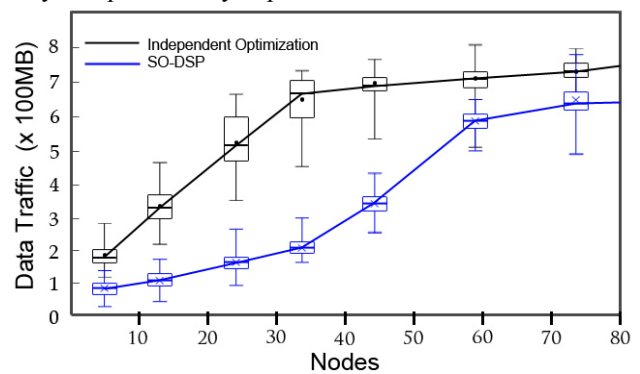


Figure 9. Data traffic in network against network size

associated with other services types that support end users. Furthermore, since SO-DSP only manages other service instances, it has minimal footprint and can be hosted on many network locations without compromising overall network performance.

Figure 7 shows a box chart of the average distances the client requests travel to access a particular service. The requests have been shown to originate from the access nodes which are part of the service provider network. This models the distribution of end users across the whole network. In general, the shorter the distance the better the observed QoS and reduced network traffic. SO-DSP management results in reducing the average distance covered by a client request to reach its respective service instance and to bring equity to all clients without compromising the required QoS. Although when migration and replication are controlled, some client requests may have to access on longer routes, but when all the clients of all services are considered, SO-DSP reduces the hop count. The distance between a client and a service instance can shrink or grow depending upon the migration/replication of a service to a new location in the network. If a service can find an optimal location where it is within reach of the end users located at different physical locations while providing acceptable QoS, it will result in a lower hop-count for all end users. SO-DSP ensures this by considering all the factors in determining which service should be placed at which particular location inside the network. Figure 8 shows the number of dropped service requests for end users during simulation for both SO-DSP

and independent service optimization approaches. Here again, the results advocate the effectiveness of SO-DSP in terms of improving the clients observable QoS parameters and overall network load.

Figure 9 shows the total data traffic in the network in case of independent optimization and SO-DSP management service. Here we observe a steep rise in the data traffic initially for independent optimization whereas a gradual increase is shown for SO-DSP instance. The rise in total data traffic with independent optimization is a result of end user's requested data as well as the data exchanged for creation of required number of replicas for different services. With SO-DSP management service, initial random placement is performed independently by each service but the subsequent migration and replication are authorized by SO-DSP instance. Since SO-DSP takes into account the overall network state and the requirements of all service types, the load on the network is more uniform.

5. Conclusion & Future Work

This paper proposed a novel network service called Self-Organized Disjoint Service Placement (*SO-DSP*) service for the management of self-organized services offered by a network. SO-DSP is a distributed service which does not require prior network knowledge and scales well to the future communication networks requirements. To prove the capabilities of *SO-DSP* further in a practical networking test bed, an in-depth evaluation of the proposed technique is required and is the topic of future work which shall be carried out in the scope of the Graduate School on Mobile Communication at the Ilmenau University of Technology [7]. The future work on *SO-DSP* includes the formal specification of *SO-DSP* service and its practical implementation on a test-bed consisting of 50 computers on the same network. Additionally, the associated cost functions will have to be developed for further analysis. The self-configuration of network entities, self-optimization, and self-healing are the main operational aspects on which the focus is laid in future communication networks. Such capabilities, when present in a network, provide an autonomous means to provide the users with the best possible services and resource utilization of the network.

References

- [1] S. Ali, A. M. Thiel, A. Diab, "Self-Organization Methodologies for Service Placement in Future Mobile Communication Networks", *In proc. Intl Conf. on Frontiers of Information Technology, FIT-09*, Dec 2009.
- [2] P. Bellavista, A. Corradi, E. Magistretti, "REDMAN: An Optimistic Replication Middleware for Read-only Resources in Dense MANETs", *International Journal on Pervasive & Mobile Computing* 1(3):279–310, Aug 2005
- [3] J. Coppens, T. Wauters, F. D. Turck, B. Dhoedt, P. Demeester, "Evaluation of Replica Placement and Retrieval Algorithms in Self-Organizing CDNs", *In Proc. IEEE Intl. Workshop on Self-Managed Systems & Services (SelfMan'05)*, May 2005
- [4] C. Frank K. Romer, "Distributed Facility Location Algorithms for Flexible Configuration of Wireless Sensor Networks", *In 3rd IEEE Intl Conf. on Distr. Computing in Sensor Systems*, Santa Fe, USA, Jun 2007
- [5] T. Furuta, M. Sasaki, F. Ishizaki, A. Suzuki, H. Miyazawa, "A New Clustering Algorithm Using Facility Location Theory for Wireless Sensor Networks", *Tech Report, Nanzan Academic Society for Mathematical Sciences and Information Engineering*, Mar 2007
- [6] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", *IEEE Transactions on Wireless Networking*. Oct 2002
- [7] International Graduate School on Mobile Communications Ilmenau University of Technology online: www.gs-mobicom.de
- [8] S. Jamin, C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt, "Constrained Mirror Placement on the Internet", *in proceedings of IEEE INFOCOM*, Mar 2001
- [9] B. Krishnamurthy, J. Wang, "On Network-aware Clustering of Web Clients," *in proceedings of ACM SIGCOM*, Aug 2000
- [10] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, A. Bestavros, "Distributed Placement of Service Facilities in Large-Scale Networks", *In 26th IEEE Conf. on Computer Communication USA*, May 2007
- [11] B. D. Lee, J. B. Weissman, "Dynamic Replica Management in the Service Grid", *In Proceedings of 10th IEEE HPDC-10*, Aug 2001
- [12] B. Li, M. J. Golin, G. F. Italiano, X. Deng, "On the Optimal Placement of Web Proxies in the Internet", *in Proceedings of IEEE INFOCOM*, Mar 1999
- [13] B. Li and K. H. Wang. "NonStop: Continuous Multimedia Streaming in Wireless Ad Hoc Networks with Node Mobility". *IEEE International Journal on Selected Areas in Communications*, 21(10) 1627–1641, Dec 2003
- [14] T. Moscibroda and R. Wattenhofer, "Facility Location: Distributed Approximation", *In 24th ACM Symposium on the Principles of Distributed Computing, Las Vegas, USA*, Jul 2005
- [15] F. Sailhan, V. Issarny, "Scalable Service Discovery for MANETs", *in IEEE International Conference on Pervasive Computing & Communications Kauai, USA*, Mar 2005
- [16] E. Yilmaz, Y. Manzano, "Surveying Formal and Practical Approaches for Optimal Placement of Replicas on the Web", <http://www.cs.fsu.edu/research/reports/TR-020701.pdf>, Apr 2002
- [17] M. Lipphardt, J. Neumann, S. Groppe, and C. Werner. "DySSCo - A Protocol for Dynamic Self-organizing Service Coverage" *In Proceedings of IWSOS '08, Vienna, Austria*, Dec. 2008
- [18] K. Herrmann. Self-Organizing Infrastructures for Ambient Services. *PhD thesis, Technische Universitat Berlin, Berlin, Germany*, July 2006
- [19] G. Wittenburg and J. Schiller, "Service Placement in Ad hoc Networks" *PhD Thesis, Freie University, Berlin Sep 2010*