

DOI: <https://dx.doi.org/10.21123/bsj.2022.6338>

Distributed Heuristic Algorithm for Migration and Replication of Self-organized Services in Future Networks

Manar AL-jabr * 

Ali Diab 

Jomana AL-Diab 

Department of Auto Control and Computers, College of Mechanical and Electrical Engineering, Al-Baath University, Homs, Syria.

*Corresponding author: albaath.homs2@gmail.com or (manar.jabr@hotmail.com)

E-mail addresses: dringalidiab@outlook.de, jomanad162@gmail.com

Received 19/5/2021, Accepted 13/10/2021, Published Online First 20/5/2022, Published 1/12/2022



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

Abstract:

Nowadays, the mobile communication networks have become a consistent part of our everyday life by transforming huge amount of data through communicating devices, that leads to new challenges. According to the Cisco Networking Index, more than 29.3 billion networked devices will be connected to the network during the year 2023. It is obvious that the existing infrastructures in current networks will not be able to support all the generated data due to the bandwidth limits, processing and transmission overhead. To cope with these issues, future mobile communication networks must achieve high requirements to reduce the amount of transferred data, decrease latency and computation costs. One of the essential challenging tasks in this subject area is the optimal self-organized service placement. In this paper a heuristic-based algorithm for service placement in future networks was presented. This algorithm achieves the ideal placement of services replicas by monitoring the load within the server and its neighborhood, choosing the node that contributes with the highest received load, and finally replicating or migrating the service to it based on specific criteria, so that the distance of requests coming from clients becomes as small as possible because of placing services within nearby locations. It was proved that our proposed algorithm achieves an improved performance by meeting the services within a shorter time, a smaller bandwidth, and thus a lower communication cost. It was compared with the traditional client-server approach and the random placement algorithm. Experimental results showed that the heuristic algorithm outperforms other approaches and meets the optimal performance with different network sizes and varying load scenarios.

Keywords: Communication Cost, Heuristic, Migration, Replication, Service Placement.

Introduction:

Nowadays, the world is experiencing a rapid growth in the size and complexity of communication networks, so the optimization and effectiveness of these networks have become an urgent requirement for network providers in recent times.

In the first decade of the twenty-first century, high-performance data networks such as 3GPP HSPA and 3GPP2HRPD were employed, with increased use of smartphones contributing significantly to the growth of data traffic within large-scale wireless networks ¹.

Fig. 1 shows The Cisco Visual Network Index (VNI) which indicates that global data traffic of future networks will increase threefold in 2022 compared to 2017 ².



Figure 1. Cisco Visual Networking Index: Forecast and Trends,2017-2022

The techniques used for distributing and replicating contents in traditional Content Distribution Networks (CDN) in terms of copying

content to sites close to clients are insufficient in recent wireless communication networks, especially in case of the large bandwidth required for these networks due to increasing demands of services by clients such as multimedia and video services³.

The expansion of innovative mobile phone services requires access technologies to enable using of available resources within a very small response time, that cannot be achieved through the current centralized systems, and this trend is expected to continue unabated and to play an important role in the 5G networks for the sake of support for both mathematically complex and time-sensitive services⁴.

Note that data traffic continued to increase significantly with the entry of the fifth generation (5G) telecommunications networks into service⁵. It is worth noting that the fifth generation is the advanced generation of communication networks capable of providing Internet of Things (IoT) requirements⁶.

In order to adapt to the growth of traffic within future networks, more complex infrastructure requirements must be implemented while meeting the quality of provided services.

With the diversity and increasing demand for services in future networks, accessing these services directly from data centers via the Internet may require a large response time due to the long transmission and reception periods (Round Trip Time (RTT)) and congestion over the WAN⁷, which gives a negative indication in the case of Time-sensitive applications such as High Quality Video Streaming, Mobile Gaming, Augmented Reality and others.

In order to reduce response time, many researchers have suggested to replicate services close to end users to the edge of the network or the cloud which adds a new hierarchical layer called (Cloudlets) or (Foglets) located between the end user and the central cloud within Mobile Edge Computing (MEC) or Fog Computing research field⁸.

Achieving these important and crucial goals requires continuous self-organization and adaptation actions in order to keep pace with ongoing changes while maintaining performance optimized. Self-organizing Networks (SON) is achieved through a range of auto-self-organization functions, and reduction of human intervention in planning, deployment, and maintenance¹.

One of the main important actions in this context is the self-organized service placement. Service placement refers to the problem of selecting the adequate nodes in the network for hosting a service. An ideal placement of service instances (replicas) in

a network minimizes the cost of serving clients by decreasing the overall network traffic, the response times and thus the communication cost.

Service can be defined as stated by Service Oriented Architectures (SOA) as a mechanism enables access to a set of capabilities by a specific pre-described interface that forces policies and permissions rules related to this service⁹. It is implemented as a software component hosted by one or multiple nodes without any prior knowledge about the requesters of this service. In communication networks, service could be considered as an abstraction of applications hosted by the BS node and requested by mobile users. Those services vary between video streaming, social gaming, navigation, augmented reality and more. So they require a large bandwidth and smaller response times which could be met by caching the associated data, such as required libraries and databases, within locations close to the end users¹⁰.

In this paper, a distributed heuristic-based algorithm for replicating and migrating of service instances within future networks depending on statistical information related to constantly changing network load and topology was presented. Our algorithm introduces a solution to a problem under Ant Colony Optimization (ACO) problems with the aim of finding the optimal location of services by means of ants carrying information about distance and load. Ants lift pheromone values within visited nodes during crossing of requests from one node to another. An agent working on a server node and monitoring the load variations within it triggers a suitable service placement decision coping with the changing conditions, taking into account the stability of the system and the overhead cost resulting from adapting the network to a new configuration with new server nodes or deactivated old ones. Experimental study and results showed that the proposed algorithm achieved an improved performance in terms of meeting the required services within a shorter time, a smaller bandwidth, and thus a lower cost compared to the traditional client-server approach and the random placement algorithm.

The rest of this paper is organized as follows: The theoretical background of service placement problem and its similarity to facility location theory were presented in section two. Section three has introduced a brief review of some prior works related to this issue. The problem statement and the proposed solution have been detailed in section four. Finally, the last two sections concluded the paper and presented future works.

Theoretical Background:

Mathematically, there are lots of problems in common between the service placement issue and facility location theory, which belongs to the field of operations research. This theory tries to solve problems of finding the optimal placement of facilities depending on mathematical models ¹¹.

Several issues have been studied in this research area, and two of them are related to service placement problem, they are (p-media) and un-capacitated facility location theory.

P-median Problem:

Summarizing the foundation introduced by Reese ¹², the p-median problem can be stated as follows: Given a network represented by a graph $G = (V, E)$, finding a set of vertices $V_p \subseteq V$ that $|V_p| = P$, where p may be either variable or fixed, and the sum of the shortest distances from vertices in V/V_p to their nearest vertex in V_p is minimized. The p-median for a graph G, is identified as a subset of vertices \hat{v}_p that satisfies the relation included in Eq.1:

$$\forall_{V_p \subseteq V} \sum_{i=1}^n w_i \cdot d(v_i, \hat{v}_p) \leq \sum_{i=1}^n w_i \cdot d(v_i, v_p) \dots\dots\dots 1$$

Where:

- $v_i \in V$ represents the vertices of the graph.
- w_i is the weight of vertex v_i .
- $d(v_i, v_j)$ indicates the weight of edge between v_i and v_j .
- $d(v_i, v_p)$ represents the shortest path between v_i and the closest vertex within V_p .

The p-median problem can be formulated as an Integer Program with an objective function as it is shown in Eq.2:

$$\min w = \sum_{ij} w_{ij} x_{ij} \dots\dots\dots 2$$

Subject to:

x_{ij} is the allocation variable equivalents to 0 or 1

$$\sum_{j=1}^{|V|} x_{ij} = 1 \quad ; j = 1, 2, \dots, |V|$$

$$\sum_{i=1}^{|V|} x_{ij} = 1 \quad ; i = 1, 2, \dots, |V|$$

$$\forall_{i, j=1 \dots n} x_{ij} \leq x_{ii}$$

$$x_{ij} \in \{0, 1\}$$

$$\sum_{i=1}^{|V|} x_{ii} = P \quad ; i = 1, 2, \dots, |V|$$

The Un-Capacitated Facility Location Problem (UFLP):

The Un-Capacitated Facility Location Problem (UFLP) was introduced by Cornuejols, Nemhauser, and Wolsey ¹³. This problem tends to decrease the cost or increase the profit resulting from satisfying the demands on a particular commodity by customers. There is, in fact, a fixed cost for deploying the services, and a transportation cost resulting from fulfilling the requested commodities by customers.

If a group of customers I requires a particular commodity from a range of services distributed among locations J . Placing the service within a location $j \in J$ entails a fixed cost f_j , while serving the customer $i \in I$ indicates a cost d_{ij} . The way to solve the UFLP, requires establishing of facilities on a subset of existing locations $\hat{j} \subseteq J$ in order to minimize the total cost, while satisfying all requests. Every Client $i \in I$ fulfills his demands from the facility j at a low cost, therefore, the whole cost of an optimal provisioning of facilities is given in Eq.3:

$$\sum_{i \in I} \min_{j \in \hat{J}} d_{ij} + \sum_{j \in \hat{J}} f_j \dots\dots\dots 3$$

A UFLP problem can be formulated mathematically as follows:

Suppose the allocation variables ϕ_{ij}, ψ_j are declared as follows:

$$\phi_j = \begin{cases} 1 & ; \\ 0 & ; \end{cases} \quad \text{if the service is available otherwise}$$

$$\phi_{ij} = \begin{cases} 1 & ; \text{if the client request of service} \\ 0 & ; \text{otherwise is met} \end{cases}$$

The Integer Program for this issue is specified in Eq.4:

$$\min w = \sum_{i \in I} \sum_{j \in J} d_{ij} \cdot \phi_{ij} + \sum_{j \in J} f_j \cdot \phi_j \dots\dots 4$$

Subject to:

$$\forall i \in I \sum_{j \in J} \phi_{ij} = 1$$

$$\forall i \in I, j \in J \rightarrow \phi_{ij} \leq \phi_j$$

$$\forall i \in I, j \in J \rightarrow \phi_{ij}, \phi_j \in \{0, 1\}$$

The solution to this problem is given in Eq.5:

$$\{j \in J \mid \phi_j = 1\} \dots\dots\dots 5$$

If the condition $\sum_{j \in J} \phi_j = P$ is added to the UFLP problem, then with a constant cost $f_i = 0$, for every facility $j \in J$, the UFLP problem corresponds to p-median one. The complexity of these problems is NP-hard, and several solutions were proposed to solve them. Depending on the definitions of the k-median and un-capacitated facility location problems, it is obvious that there are excessive matches to the service placement problem. More specifically, service placement problem can be considered as an application of facility location theory to service provisioning in communication networks¹⁴.

Service placement techniques can be categorized into centralized, distributed and hybrid approaches including a wide range of algorithms¹⁵. Centralized approaches depend on central controlling entity contains information about the whole system. It is easy to manage and apply these techniques, but they suffer from single point of failure. Distributed service placement techniques are more reliable and flexible. They depend on local neighborhood information in tacking placement decisions, so they are scalable and applicable to future networks. Hybrid approaches use both centralized and distributed techniques. There are a wide range of algorithms fall under the former techniques.

Relevant Prior Work:

In 2008 Gramoli, Kermarrec and Atlantique, proposed SONDE (Self-Organizing Network Density) algorithm to find the optimal number and locations of servers among the network, by periodically verifying server existence within a node vicinity (h-local neighborhood) and turning the node state from client to server if the last is not found in its neighborhood, while turning the server node into a client if it finds oldest one within its vicinity¹⁶. The number of providers increased (or decreased) if the server node is overloaded (or under loaded). The authors approved that the number of servers stabilizes after a short period of time, and the service requests are met with a low latency and an optimized throughput.

In 2016, Poularakis and Tassiulas stated that traditional methods applied to expand network bandwidth to meet the continuously growing demands for mobile data are expensive, and the alternative approach to overcome this challenge is to replicate objects to edges close to users, so that the response time will be minimized and the throughput will be increased. They showed that this problem is NP-hard and related to knapsack and facility location problems¹⁷.

In 2016, Sahoo, Salahuddin, et al. have studied

the server placement problem in both traditional content distribution networks and emerging cloud-based CDN as well as virtual network function (VNF) based CDN. Researchers have pointed to the proposed multi-objectives algorithms taking into account multiple criteria such as (response time RTT (Round Trip Time), latency, hop-count, link quality, cost, number of servers, number of reconfigurations and network traffic).

They demonstrated that the most appropriate model for solving this problem was the Integer Programming one, and most algorithms relied at 42% on the greedy algorithm, and others based on the heuristic methods at percentage of 6%. The paper summarized the possible guidelines for future visions in the research area of service placement¹⁸.

In 2017, Skarlat, Nardelli, Schulte, et al. introduced a framework for conceptual fog computing. They modeled the service placement problem for IoT applications as an optimization problem, and presented a heuristic solution by genetic algorithm. Their results proved that the communication delays were minimized and resources usage was improved. The optimization problem was solved by different approaches, namely the exact optimization method and its approximation through a greedy first fit heuristic and a genetic algorithm¹⁹.

Albu-Salih and Hosseini Seno in 2018, proposed a new scheme for energy efficient data collection with a deadline time for the Internet of things (IoT) using the Unmanned Aerial Vehicles (UAV). They provided a mixed integer linear programming model (MILP) in order to find an optimal solution to this problem, and then they used a heuristic to solve the time complexity problem. Simulation was performed to compare the performance of the MILP optimization method and the greedy method in different scenarios. The results obtained in the simulation results indicated the optimal performance of the proposed scheme in terms of energy consumption and the number of used UAVs²⁰.

In 2019, Selimi, Cerdà-Alabern, Freitag, et al. proposed a fast heuristic algorithm called BASP (Bandwidth and Availability-aware Service Placement) in community networks (CNs) that react quickly to changing conditions based on leverage state information in tacking placement decision. They compared their algorithm with random placement one in the biggest CN world-wide (Guifi.net) and showed that their algorithm outperforms the random placement by 2x gain in bandwidth²¹.

In 2019, Mueller-Bady, Kappes, Medina-Bulo et al. developed a heuristic technique based on

effective hybrid search using the high arithmetic capability of the greedy local search method to find the optimal locations and numbers of the monitoring nodes among communication networks, thus achieving an optimized performance while adapting to changes in the network topology ²². They stated that this method works well when reacting to network changing conditions and when deploying millions of nodes compared to various evolutionary algorithms (EA).

Donassolo, Fajjari, Legrand, et al. in 2019, have introduced micro services placement problem in IoT as an optimization problem. They proposed an Optimized Fog Service Provisioning strategy(O-FSP) and proved the acceptance of this solution with respect to the cost and consumed resources compared with some traditional aspects such as Best-fit algorithm ²³.

Problem Statement and Proposed Solution:

For a representative graph of the network $G=(V, E)$ where vertices V corresponds to nodes in future networks, while the edges E between them represent the links, so that $E \subseteq V * V$ ²⁴.

1. For x_{ij} denotes a decision variable that indicates the allocation of vertex v_i to vertex v_j :

$$x_{ij} = \begin{cases} 1 & \text{If } v_i \text{ was assigned to } v_j \\ 0 & \text{Otherwise} \end{cases}$$

2. Each service type S_i has number of replicas due to changing conditions of load and topology. When a client issues requests for a service S_i , the look up or service discovery process returns the nearest server node as indicated in Eq.6:

$$\eta(s, c) = \arg \min_{h \in H} d_{h,c} \dots\dots\dots 6$$

Where $d_{h,c}$ is the distance between client c and server host h .

3. The cost function with respect to the client demand bandwidth and the distance is represented in Eq.7:

$$C_{clients}(s, h) = \sum_{c \in C_{s,h}} d_{\eta(s,c)} \delta(s, c) \dots\dots 7$$

Where:

- $d_{\eta(s,c)}$ is the shortest virtual distance between client and server in terms of hop count.
- $\delta(s, c)$ is the bandwidth of client demand.

It is obvious that reducing the virtual distance between clients and servers plays vital role in minimizing the overall communication cost.

4. No global knowledge of the network topology is available.
5. Finding the optimal placement of services in the network is NP-hard problem.

To solve this problem, a heuristic-based distributed algorithm has been proposed tacking into account statistical information relevant to the load and network topology in order to find the optimal number and locations of service replicas, and thus minimizing the communication cost and retaining quality of service.

Cost Model:

Given a set of network nodes N . Each one is capable of hosting a set of services M with a maximum capacity S_i byte. Every client n_i requests the service of size b_m bytes at a probability P_m and rate λ_i , where $m \in \{1,2,\dots,M\}$. The allocation variable is stated as follows:

$$x_{im} = \begin{cases} 1 & \text{if service } m \text{ is available in } n_i \\ 0 & \text{otherwise} \end{cases}$$

Within a constraint:

$$\sum_{m=1}^M b_m x_{im} \leq S_i \quad i = 1, \dots, N$$

The object of this integer problem is to serve a client from the closest server i.e. the number of hops crossed by the request is as small as possible. Suppose that each service is available on origin server O_m . X_o refers to the availability of origin copy of a service within the server node. The average number of hop counts the request travers from a client to a server is given Eq.8:

$$C_i(X) = \sum_{m=1}^M P_m d_{im}(X) \dots\dots\dots 8$$

Where:

$d_{im}(X)$ is the smallest distance crossed to get the service m from a node n_i when the placement configuration X takes a place.

The service is available either within the origin node or within another node hosting a replica of a service, but the client request is always forwarded to the closest server node, then with a total request rate $\mu = \sum_i \lambda_i$, the average number of hops for all nodes is given in the following:

$$C(X) = \frac{1}{\mu} \sum_{i=1}^N \lambda_i C_i(X) \\ = \frac{1}{\mu} \sum_{i=1}^N \sum_{m=1}^M \lambda_i P_m d_{im}(X)$$

$$= \sum_{i=1}^N \sum_{m=1}^M S_{im} d_{im}(X)$$

$$\text{Where: } S_{im} = \frac{\lambda_i P_m}{\mu}$$

And the placement X is taken under the constraint:

$$\sum_{m=1}^M b_m x_{im} \leq S_i \quad i = 1, \dots, I$$

For a large number of server nodes and services, it is difficult to find the optimal solution to this problem, and it is NP-complete.

NP-completeness can be proved by formulating this problem as an optimal integer program, and then proving that the knapsack problem is reduced to it. Since a service placement problem is NP-hard, finding the optimal solution is infeasible, and could be achieved by heuristic methods.

Heuristic Algorithm:

In this paper, a proposed algorithm based on ant behavior was presented. Ant Colony Optimization (ACO) is popular meta-heuristic evolving algorithms that are used to generate solutions near to optimal one for a large diversity of optimization problems²⁵.

Similar to Ant-Colony Optimization problems, communications between nodes are achieved by lifting cues in the environment as an alternative to messages exchange.

The overall process is divided into two phases:

- **Initialization phase:** In this phase, the server nodes notify others about their existence. When the client node receives these messages, it stores the service information (service ID, location, service priority, data rate, packet size, number of packets) within its own memory in order to look it up during service discovery phase before issuing requests. It is possible to implement the service discovery component within specific nodes spread among multiple network partitions, then the demand for services starts with issuing service discovery request to a neighbor node responsible for sending a reply with the id of suitable server hosting the requested service.
- **Operation phase:** The clients start issuing requests of services after selecting the closest server node in the service discovery phase. When the autonomous server node starts, it employs an agent responsible for monitoring the load within it, and then when the server is overloaded, it selects the appropriate node to which the service will be replicated or migrated.

The following section details how the algorithm works:

Our proposal indicates a bio-inspired ant based algorithm that depends on load statistical information saved in pheromone tables within the nodes. The pheromone value in this case refers to the traffic size (in terms of bytes) generated by the clients and crossing the node i.e. ants travel from one node to another depositing the amount of load within it. It can be said as an imitation of ant algorithm²⁶, that the service denotes a source of food while the requests generated by clients represent the ants. Whenever a new packet crosses the node, a pheromone value of its size is left within it. The pheromone value is updated due to Eq.9:

$$P_s(t+1) = \frac{P_s(t) + \delta p(h)}{1 + \delta p(h)} \quad \dots\dots\dots 9$$

Where $\delta p(h)$ is the change in pheromone value, and $P_s(t)$ is the size of traffic at time t.

When the server node is overloaded for a period of time, i.e. the number of requests (and consequently the traffic byte size) bypass a specific threshold, it automatically decides to replicate or migrate a service to a new node in case of sufficient resources to host the service are available within it. The selected node is the last hop that contributes the maximum amount of load within a server neighborhood.

The decision of migrating the service or replicating it, is subject to the following rules:

- **Migration Rule:** The migration rule says: migrate the service to a node n, if the number of requests (load) coming from this node is greater than the sum of all requests coming from all other nodes belonging to server neighborhood.
- **Replication Rule:** Given a replication radius η , the replication rule says: replicate a service to a neighbor node n if this node averages a request path length of at least η and contributes the highest request flow.

The function used to calculate the communication cost between clients and servers depends on the load (in terms of bytes) and the crossed distance (in terms of hop count) (Table 1). The Pseudo Code of the proposed algorithm is presented in the next section (Table 2).

Table 1. Communication Cost between Clients and Servers

Function 1. Calculate Communication Cost

```

Calculate_Cost (C, nodecurr)
Require: C(List of clients nodes requesting the
service), nodecurr (Current node hosting service)
cost ← 0
for u ∈ C do
    π(u) ← get-client_load(u)
    h(u) ← get-hop-count (u)
    cost ← cost + π(u) * h(u)
end for
return cost

```

Table 2. The Pseudo-Code of the proposed Algorithm

Algorithm 1. Heuristic Algorithm

Require: The network nodes N, the neighbor list of each node, the clients demands for each service s by node n, the size and the synchronization ratio of the service, hop counts(distances) between nodes.
Output: Optimal service configuration(placement)

```

for u ∈ N do
    if (provider (u)) do
        if ( IsOverld(u) and period > μ )
            Costcurrent ← Calculate_Cost (C, u)
            R ← Get_Replica-Node()
            If (R ≠ ∞ )
                Costnew ← Calculate_Cost (CR, R) + RepCost +
SyncCost
                If (Costnew < Costcurrent )
                    Replicate (u,R)
                End if
            Else
                G ← Get_Migration-Node()
                If (G ≠ ∞ )
                    Costnew ← Calculate_Cost (CG, G) +
MigCost
                    If (Costnew < Costcurrent )
                        Migrate (u,G)
                        Deactivate(u)
                    End if
                End if
            Else if( IsUndrld(u) and period > μ)
                Deactivate(u)
            End if
        end if
    end for
end for

```

The proposed algorithm requires the following information as input. First, the neighbor list of each node which is calculated periodically during the simulation and stored in a vector within a node. Second, the clients demand for each service and hop counts (distances) between nodes are updated by the server node as a new request is received. Finally, the synchronization ratio of the service is supposed to be as a percentage of the total traffic.

Each packet comes from the client brings the

virtual distance value (hop counts) between the client and the server nodes, so the communication overhead caused by one packet is given by multiplying the packet size value with this distance value (hop count). As a result, the overall cost is given by the sum of all cost values related to packets in the request. The decision to replicate or migrate is made automatically by the node based on the observed statistical information related to clients' loads and network topology. This information is given in the following:

- Request count.
- Distance (Hop count).
- Address of last hop.
- Data rate(kb/s).
- Packet size(byte)

Additional amount of information is taken into account based on statistics related to a subset of last-hop nodes forwarding requests to servers. One of these nodes is elected to host a service replica. The last-hop nodes statistics are listed in the following:

- Packet count: refers to total number of packets traversing the node.
- Total bytes: stands for the total size of traffic traversing the node in bytes.

When the server node is overloaded for a period of time, it tries to find a replication target node. If the last is found depending on the rule described above, the agent in the server calculates the new cost resulting from deploying the service within the new node in addition to replication and synchronization costs. The placement decision is taken into account if the new cost is less than the old one. If the replication process is not accomplished, the origin node tries to migrate the service to another node if the new resulting cost with that node in addition to migration cost is less than the old cost.

When the decision of replicating or migrating is made by the server node, the clients are notified about the new location of the service.

If the server node is under loaded for a specific period of time, it is shut down and removed from the clients' memories.

Materials and Methods:

The proposed algorithm was implemented in Ns-3 Network Simulator. Ns3 is a simulator for discrete-event network and Internet systems written in multiple languages (C++/OTcl)²⁷. Ns3 is:

- Open source licensing (GNU GPLv2) and development model.
- Python scripts or C++ programs.
- Satisfies configuration with real systems (device driver interfaces, sockets) and also

association with (pcap traces, ns-2 mobility scripts).

- Test bed assimilation is an importance.
- Modular, and documented in core.

In our evaluation, the advantages of our heuristic service placement algorithm over the traditional client-server approach and the random placement algorithm was quantified. The network was simulated of different sizes (in terms of number of nodes) and under different load scenarios.

Table .3 shows a set of simulation parameters used in the experiments:

Table 3. Parameters for Simulations.

Parameter	Value
Area	1000 × 1000 m ²
Network Topology	Nodes are Randomly placed within a grid
Mobility Model	Random Way Point
Position of nodes	Randomly placed
Number of Nodes	(25,40,50,80)
Number of Servers	10% of total initially
Clients locations and demands	Random
Number of Services types with different loads (requests)	5
Message Length	1040 bytes
Default Data Rate	1Kbps
Simulation Time	30 min.

Results and Discussion:

The network was simulated for different sizes of nodes (25,40,50,80) that are randomly placed within a grid and for different load scenarios. Each node has a radio link to each of its direct neighbor nodes. It was supposed that the network offer 5 different types of services distributed initially within 10% of total nodes. A single service instance of each service type is started on randomly selected nodes. The network was simulated for 30 minutes, and obtained the results from aggregated 15 runs. After a configuration time all clients nodes issue requests with specific data rates at 1Kbps and maximum number of packets. This number of packets is not unified for all clients. It was supposed that for each client node the number of requests equals to the remainder of dividing the node ID by 10. The placement algorithm is triggered on a node once it starts playing its role as a server. When a migration or a replication decision is made, a specific amount of bytes is transferred from source to destination. The size of transferred service state was assumed to be 50KB, and in case of replication there is a synchronization traffic of 20% of total traffic. So that the cost of synchronization was calculated by multiplying this percentage by the cost of

communication between replicas which in turn stated as the product of distances between replicas with the demands traffic.

In our experimental study, the advantages of our placement algorithm over the traditional client-server approach and the random placement algorithm was measured. The results showed that the average distance of service requests between client nodes and server ones decreases with the proposed algorithm due to creating of a proper number of service replicas. Fig. 2 represents graphically the average hop count travelled by client requests in case of applying our heuristic placement algorithm, random placement algorithm, and traditional client-server model with different network sizes (different number of nodes).

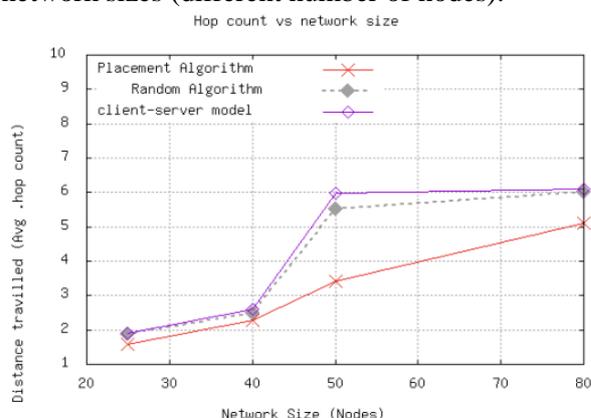


Figure 2. Average travelled hop count against network size.

As it is shown in Fig. 2, it is obvious that our placement algorithm outperforms the two other approaches as it reduces the distances travelled (hop count) by clients requests to reach server nodes. Fig. 3 and Fig. 4 show that our service placement algorithm also minimizes the round-trip time compared to other approaches for different network sizes and loads.

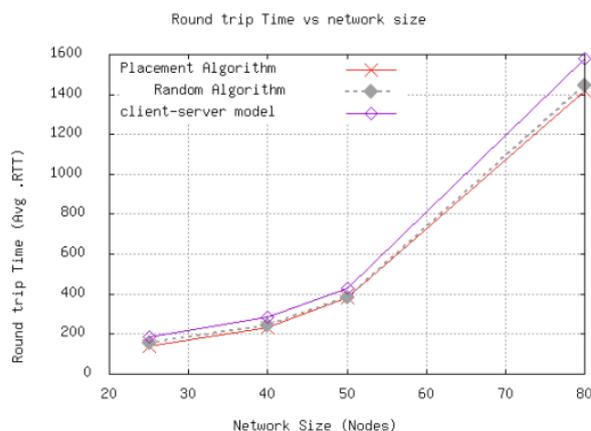


Figure 3. Average Round Trip Time against different network sizes.

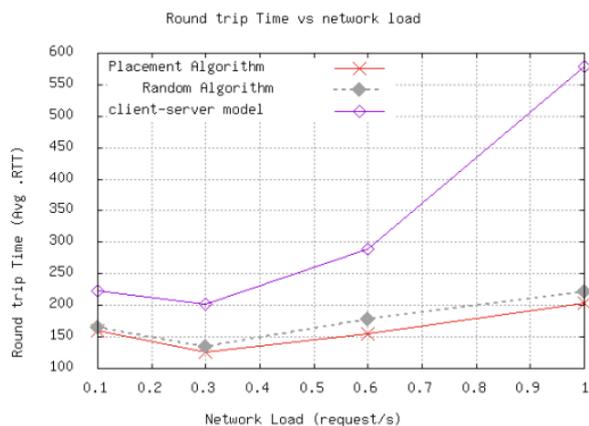


Figure 4. Average Round Trip Time against different network loads.

The communication cost was also evaluated between clients and servers in terms of consumed bandwidth and hop count, and then represented graphically in Fig. 5 for different network sizes and in Fig. 6 for different network loads.

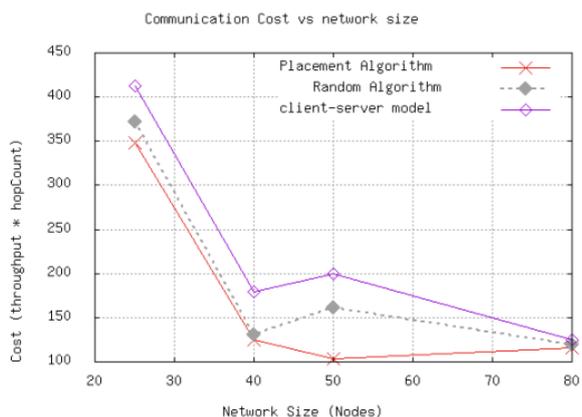


Figure 5. Communication Cost against different network sizes.

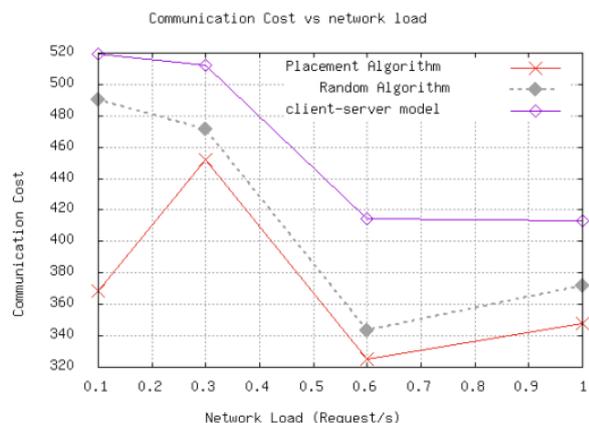


Figure 6. Communication Cost against different network loads.

In the following, Fig. 7, represents the Packet Delivery Rate between clients and servers for

different network sizes. The heuristic algorithm achieves delivery of packets between sources and destinations at rates higher than the two other approaches. Hence, the result can then be summarized as our proposed algorithm fulfills a quality of service requirements and reliability over the whole network, so services are constantly available and an optimized performance was successfully achieved with minimized cost.

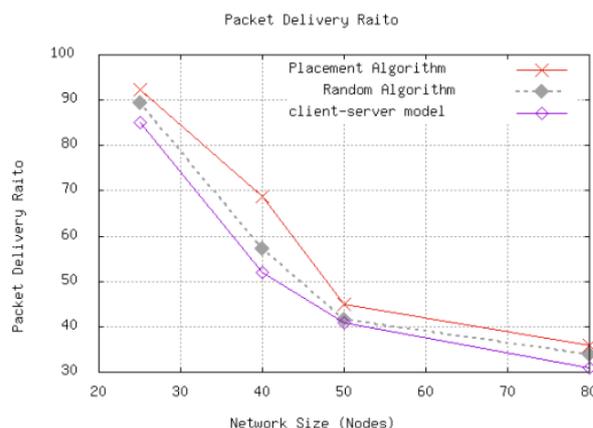


Figure 7. Packet Delivery Rate against different network sizes.

Finally, the efficiency of the previous approaches was estimated taking into account the time cost. It was shown that the client-server model approach has constant time complexity $O(1)$, while the random algorithm takes linear time $O(N)$ to be executed. The proposed heuristic algorithm takes time complexity of $O(N*M)$ which is multi-linear. Finally, the obtained results conclude that the proposed service placement algorithm decreases the overall communication cost and improves the service quality (via minimizing RTT).

Conclusion:

In this paper, a bio-inspired algorithm for placement of services in future networks was developed. This algorithm is based on ant behavior depending on statistical information of load and continuing changing topology of the network. When the server node is overloaded a decision of replicating or migrating services to neighbor nodes is taken into account minimizing the hop counts travelled by clients requests, round trip time, overall network traffic and hence the cost of communication between clients and servers. The performance of our algorithm was evaluated in Ns-3 network simulator and compared with the random placement algorithm and the traditional client-server approach. Experimental results showed that the new algorithm outperforms other approaches and satisfies packet delivery at higher rates, with

minimized communication cost, so that the quality of service is met and the network became more reliable and safe against failures due to replicating or migrating of services to other server nodes.

The simulation results proved that our algorithm satisfies service provisioning in future networks with an optimized performance and acceptable level of communication cost.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in Al-Baath University.

Authors' contributions statement:

A D and J AD conceived of the presented idea. MA-J designed, developed the theory and performed the computations. M A-J, A D and J A-D verified the analytical methods. A D and J AD encouraged M A-J to investigate Algorithms and compare different approaches related to service problem in future networks. A D and J AD supervised the findings of this work.

All authors read the manuscript carefully, discussed the results, contributed to the final manuscript.

References:

1. Ramiro J, Hamied K, editors. Self-organizing networks, self-planning, self-optimization, and self-healing for GSM, UMTS and LTE. USA: Wiley; 2012. 318 p.
2. Barnett Th, Jain Sh, Andra U, Khurana T. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022. APJC Cisco Knowledge Network (CKN).2019 Feb: 33p.
3. Hachem J, Karamchandani N, Diggavi S. Content caching and delivery over heterogeneous wireless networks. Proc IEEE INFOCOM Online. 2015 Aug 24, 756-764, DOI: 10.1109/INFOCOM.2015 May.7218445 .
4. Poularakis K, Llorca J, Tulino A. M, Taylor I, Tassiulas L. Joint Service Placement and Request Routing in Multi-Cell Mobile Edge Computing Networks. Proc IEEE INFOCOM Online. 2019 June 17, 10-18, DOI: 10.1109/INFOCOM.2019.8737385.
5. Gupta A, Jha R K. A Survey of 5G Network: Architecture and Emerging Technologies. IEEE Access. 2015 Jul 28; 3:1206-1232, doi: 10.1109/ACCESS.2015.2461602.
6. Chiang M, Zhang T. Fog and IoT: An overview of research opportunities. IEEE Internet Things J. 2016 Dec; 3(6): 854 – 864.
7. Yang L, Cao J, Liang G, Han, X. Cost aware Service Placement and Load Dispatching in Mobile Cloud Systems. IEEE Trans Comput. 2016 May 01; 65(5): 1440 – 1452, <https://doi.org/10.1109/TC.2015.2435781>
8. Abbas N, Zhang Y, Taherkordi A, Skeie T. Mobile Edge Computing: A Survey. IEEE Internet Things J. 2018 Feb; 5(1): 450-465, doi: 10.1109/JIOT.2017.2750180.
9. Mackenzie C M, Laskey K, McCabe F, Brown P F, Metz R B. Reference Model for Service Oriented Architecture 1.0. OASIS standard. 2006 Aug [cited 2018 Nov 19]; 31p. Available from: <http://docs.oasis-open.org/soa-rm/v1.0/>
10. Xu J, Chen L, Zhou P. Joint Service Caching and Task Offloading for Mobile Edge Computing in Dense Networks. Proc IEEE Infocom Online. 2018 Oct 11: 207-215, DOI: 10.1109/INFOCOM.2018.8485977.
11. Gadegaard S, L. Discrete Location Problems: Theory, algorithms, and extensions to multiple objectives. A PhD Dissertation. Denmark: Department of Economics and Business Economics, Aarhus University; 2016.
12. Reese J. Solution Methods for the p-Median Problem: An Annotated Bibliography. Networks (N Y). 2006 August 01; 48(3):125–142p, DOI: <https://doi.org/10.1002/net.20128>
13. Corneújols G, Nemhauser G L, Wolsey L A. The Uncapacitated Facility Location Problem. In: Mirchandani B. P, Francis R. L, editors. Discrete Location Theory..USA:Wiley-Interscience;1990. 99th ed ,576: 1-8.
14. Laoutaris N, Smaragdakis G, Oikonomou K, Stavrakakis I, Bestavros A. Distributed Placement of Service Facilities in Large-Scale Networks. Proc IEEE Infocom Online. 2007 May: 2144-2152, DOI: 10.1109/INFOCOM.2007.248.
15. Ali S, Mitschele-Thiel A, Diab A, Rasheed A. A Survey of Services Placement Mechanisms for Future Mobile Communication Networks. Association for Computing Machinery, Proceedings of the 8th International Conference on Frontiers of Information Technology, ACM-BCB 2010, FIT '10. 2010 Dec: 1-5, DOI: <https://doi.org/10.1145/1943628.1943667>.
16. Gramoli V, Kermarrec A, Merrer E L, Neveux D. SONDE, a Self-Organizing Object Deployment Algorithm in Large-Scale Dynamic Systems. In: 2008 Seventh European Dependable Computing Conference. IEEE, 2008:157-166, DOI: 10.1109/EDCC-7.2008.17.
17. Poularakis K, Tassiulas L. Cooperation and information replication in wireless networks. Philos Trans A Math Phys Eng Sci [Internet]. 2016 [cited 2020 Dec 02]; 374(2062): 14p. Available from: <https://doi.org/10.1098/rsta.2015.0123>
18. Sahoo J, Salahuddin M A, Glitho R, Elbiaze H, Ajib W. A Survey on Replica Server Placement Algorithms for Content Delivery Networks. IEEE

- Commun Surv. 2017; 19(2). 1002-1026, DOI: 10.1109/COMST.2016.2626384.
19. Skarlat O, Nardelli M, Schulte S, Borkowski M, Leitner P. Optimized IoT service placement in the fog. Springer London: Serv Oriented Comput. Appl.11(4). 2017Oct.04, 427-443. DOI: <https://doi.org/10.1007/s11761-017-0219-8>
20. Albu-Salih, A T, Hosseini Seno S A. Optimal UAV Deployment for Data Collection in Deadline-based IoT Applications. Baghdad Sci J. 2018Dec 9; 15(4): 0484.
21. Selimi M, Cerdà-Alabern L, Freitag F, Veiga L, Sathiaselan A, Crowcroft J. A Lightweight Service Placement Approach for Community Network Micro-Clouds. Springer Netherlands: J Grid Comput. 2019 March28; 17(1): 169-189, DOI: <https://doi.org/10.1007/s10723-018-9437-3>
22. Mueller-Bady R, Kappes M, Medina-Bulo I, Palomolozano F. An evolutionary hybrid search heuristic for monitor placement in communication networks. Springer Netherlands: J Heuristics. 2019May02; 25(6): 861-899. DOI: <https://doi.org/10.1007/s10732-019-09414-z>
23. Donassolo B, Fajjari I, Legrand A, Mertikopoulos, P. Fog Based Framework for IoT Service Provisioning. 16th IEEE Annu. Consum. Commun. Netw. Conf. 2019 Feb28: 1-6, DOI: 10.1109/CCNC.2019.8651835.
24. Diestel R. Graph Theory (Graduate Texts in Mathematics, 173), 5th ed. Berlin, Heidelberg: Springer. Extremal Graph Theory. 2017. Chap7: 173-207.
25. Ali F, Jawad R. Using Evolving Algorithms to Cryptanalysis Nonlinear Cryptosystems. Baghdad Sci J. 2020June06; 17(2): 1-7, DOI: [http://dx.doi.org/10.21123/bsj.2020.17.2\(SI\).0682](http://dx.doi.org/10.21123/bsj.2020.17.2(SI).0682)
26. Bonabeau E, Dorigo M, Theraulaz G. Swarm Intelligence: From Natural to Artificial Systems. New York, NY: Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity; 1999. Chap 2, Ant Foraging Behavior, Combinatorial Optimization, and Routing in Communications Networks, 37P.
27. Carneiro G. NS-3: Network simulator 3. In: UTM Lab Meeting April. 2010: 4-5.

خوارزمية استدلالية موزعة لتجهيز ونسخ الخدمات ذاتية التأقلم في الشبكات المستقبلية

جمانة الدياب

علي ذياب

منار الجبر

قسم التحكم الآلي والحواسيب، كلية الهندسة الميكانيكية والكهربائية، جامعة البعث، حمص، سورية.

الخلاصة:

أصبحت شبكات الاتصالات المحمولة في الوقت الحاضر جزءاً متأسلاً في حياتنا اليومية من خلال الكميات الهائلة من البيانات التي يتم تنقلها عبر أجهزة الاتصال، مما يقود إلى تحديات جديدة. وسيتم حسب مؤشر سيسكو للشبكات، توصيل أكثر من 29.3 بليون جهاز عبر الشبكة خلال العام 2023. من الواضح أن البنى التحتية الموجودة في الشبكات الحالية لن تكون قادرة على دعم جميع البيانات التي يتم تبادلها بسبب عرض الحزمة المحدود وكلفة عمليات الإرسال والمعالجة. ومن أجل التعامل مع هذه المشكلات، يجب أن تحقق شبكات الاتصالات المحمولة المستقبلية متطلبات عالية من أجل إنقاص كمية البيانات المنقولة وتقليل زمن الوصول وكلفة عمليات المعالجة. تتمثل إحدى التحديات العلمية الهامة ضمن هذا السياق في التوزيع المثالي للخدمات ذاتية التأقلم. تم في هذه الورقة البحثية تقديم خوارزمية استدلالية لتوزيع الخدمات في الشبكات المستقبلية. تحقق هذه الخوارزمية التوزيع المثالي لنسخ الخدمات من خلال مراقبة الحمل داخل عقدة المخدم وجوارها، واختيار العقدة التي يتم تلقي الحمل الأكبر منها، ونسخ الخدمة أو تهجيرها إليها بناءً على معايير محددة، فتصبح بالتالي المسافة التي تعبرها الطلبات الواردة من العقد الزبائن صغيرة قدر الإمكان بسبب توزيع الخدمات في مواقع قريبة منها. تم الإثبات أن الخوارزمية المقترحة من قبلنا تحقق أداءً محسناً من ناحية تلبية الخدمات خلال زمن أقصر، وعرض حزمة أصغر وبالتالي كلفة اتصال أقل. أجريت مقارنة بين هذه الخوارزمية وكل من نموذج الزبون-مخدم التقليدي وخوارزمية التوزيع العشوائي. أثبتت النتائج التجريبية أن الخوارزمية الاستدلالية تتفوق على الطرق الأخرى وتحقق الأداء الأمثل من أجل شبكات بأحجام مختلفة وسيناريوهات بأحجام متنوعة.

الكلمات المفتاحية: كلفة الاتصال، الاستدلال، التهجير، النسخ، توزيع الخدمات.