EFFICIENT SOCIAL-AWARE SERVICE PLACEMENT IN DISTRIBUTED NETWORKS

Raji Rajasekharan,

M. Tech Scholar, Dept of Computer Science,
Sarabhai Institute of Science and Technology,
Vellanad, Thiruvananthapuram, India

ABSTRACT

Service deployment deals with the problem of selecting which node in a network is most suitable for hosting a service and that responds to queries from other nodes. Optimal placement of service facilities reduces network traffic and improves connectivity between clients and servers. Here it deals with the movement of service facility between neighbour nodes in a way that the cost of service provision is reduced and the service facility reaches the optimal location and remains there as long as the environment does not change, and as network condition changes the migration process is resumed automatically, Thus naturally responding to network dynamicity under certain conditions.The paper focus to bring the service provision points close to the demand in order to minimize communication cost of provided service.

Keywords-Service deployment, autonomic network, service migration.

I. INTRODUCTION

A.Background Distributed system is a collection of independent computers that appears to its users as a single coherent system. It is a software system in which components placed on networked computers communicate and coordinate their actions through messages. The components interact with each other in order to achieve a common goal. The system runs on a collection of computers that do not have shared memory, but looks like a single system to its users.It is considered as a network of processes where the edges are

communication channel and the nodes are processes.One important characteristic is that the existence of various computers and the methods by which they communicate are mostly hidden from users. The same is with the internal organization of the system. Another important characteristic is that applications and users can interact with a distributed system in a uniform and consistent way, regardless of where and when interaction takes place. Distributed systems should also be relatively easy to scale or expand. This characteristic is a consequence of having independent computers,at the same time, hiding how these systems actually take part as a whole system. A distributed system will normally be continuously available, even though some parts may be temporarily out of order. Users and applications should not notice that parts are being fixed or replaced, or that new parts are added to serve more users or applications .

To support heterogeneous computers and networks still offering a single-system view, distributed systems are organized by means of a layer of software-that is, logically placed between a higher-level layer consisting of users and applications, and a layer underneath consisting of operating systems and basic communication facilities, such a distributed system is known as middleware.

There are many different types of distributed systems and many challenges to overcome in successfully designing one. The main goal of a distributed system is to connect users and resources in a transparent, open, and scalable way. An important goal of a distributed system is to hide the fact that its processes and resources are physically distributed across multiple computers. A distributed system that is able to present itself to users and applications as if it were only a single computer system is said to be transparent. Distributed systems in which resources can be moved without knowing how these resources are accessed are said to provide **Migration transparency**. The situation is even stronger in which resources can be relocated while they are being accessed without the user or application noticing anything. In such cases, the system is said to support **Relocation transparency**. An example of relocation transparency is when mobile users can continue to use their wireless laptops while moving from place to place without ever being temporarily disconnected.

In distributed systems, replication is an important factor. For example, resources may be replicated to increase availability or to improve performance by placing a copy close to the place where it is accessed. **Replication transparency** deals with hiding the fact that several copies of a resource exist. All replicas should have the same name inorder to hide replication. A system that supports replication transparency should generally support **Location transparency** as well,otherwise it would not be possible to refer to replicas at different locations.

B. Service Deployment in Distributed Systems

Service placement is a key problem in communication networks as it determines how efficiently the user service demands are supported [5]. This problem has been traditionally approached through the formulation and resolution of large optimization problems requiring global knowledge and a continuous recalculation of the solution in case of network changes. Such approaches are not suitable for large-scale and dynamic network environments.

The service placement problem is encountered in various networks such as transportation networks, supply networks and communication networks. The globalization of the Internet and the proliferation of services and service demands have necessitated the careful selection of the location of the service[7]. The goal is to bring the service provision points (referred to here as service nodes) close to the demand in order to minimize communication resource consumption and enhance the Quality of Service of the provided service. Due to the recent technological changes (e.g., the introduction of powerful machines, the service proliferation, the generalization of high-power computing) the traditional problem of placing relatively few big services in one of the few (powerful) potential service provider facilities (big network elements) is increasingly being transformed into a problem of placing numerous services to one of the numerous potential service providers (network elements and possibly service producers).

In traditional networks service provision is typically the responsibility of the (sub-) network owner or a well defined entity that owns or leases part of the needed infrastructure and sometimes enters in agreement with network service providers. In such networks the location of the service provision is dictated by ownership limitations[9]. The globalization of the Internet and the expansion of the service demand profiles have necessitated the careful selection of the location of the service, as well as the replication of the service provision points. The objective being, to bring the service provision points (referred to here as service nodes) close to the demand in order to minimize communication resource consumption and enhance the Quality of Service (QoS) of the provided service.

The problem of service placement has received some attention in the aforementioned traditional networking environment, for example, in the context of content placement and replication in Content Distribution Networks. This problem is typically addressed by invoking approaches that do not scale with the number of services and network nodes, typically rely on some global information knowledge in order to provide for a solution under given (static) conditions and cannot inherently cope with dynamic environments. As indicated in the next paragraphs, the service generation and provision landscape and the supporting networking infrastructure are changing drastically in a way that the traditional approach to service placement is non-scalable.

The first change has to do with the proliferation and “miniaturization” of the services produced by networked nodes[9]. The emerging “long-tail” relation between percentage of content (service) produced by a certain content (service) producer reveals the fact that network services proliferate in number and type and that most of these services are “small” (i.e., easily produced by small networked nodes).In addition, the technology appears to be mature to consider service personalization and autonomic service composition which is expected to further enhance the "miniaturization" and proliferation of the network services. The second change has to do with the proliferation and "miniaturization" or the network elements, as well as network users. The term "miniaturization" may be used here to capture the fact that the traditionally heavy network elements (routers) are increasingly being supplemented by lighter network elements that are contributed by (until recently) traditional network users; these users are becoming powerful enough to engage in ad hoc networking and contribute to the networking infrastructure[3]. For example, the increasing contribution to the networking infrastructure of numerous small (traditionally) user-nodes is already materializing in lastmile networking and is expected to dominate soon (e.g., home owner based WLAN network service access). In addition, numerous new small appliances are increasingly being networked, contributing to the proliferation and miniaturization of the network users. The increasing proliferation of network infrastructure and its increasing autonomicity and ownership diversification are the main reasons for which a new network architecture is needed to organize and address the efficiency and complexity issues associated with it.

The main point from the above discussion is the proliferation of services and network nodes that calls for approaches that scale well with the numbers;"miniaturization" contributes to the proliferation, as well as calls for approaches that should be distributed and relatively light. Consequently, the traditional problem of placing relatively few big services to one of the few (powerful) potential service provider facilities (big network elements) is increasingly being transformed into a problem of placing numerous services to one of the numerous potential service providers (network elements and possibly service producers).

C. Content Distribution Network

The Internet has become much more than just a communication infrastructure, up to the point that some authors have defined the global network as a "platform for business and society". This change poses new requirements on the Internet itself, which was not specifically designed to perform content distribution. In order to tackle this issue, Content-Distribution Networks (CDNs), such as AKAMAI, have become a vital layer in the architecture of any content provider as they make it possible to distribute content in today's IP Internet in an efficient way[20]. The driving principle of CDNs is that content requests are not directly served by the origin server owned by the content provider, but they are instead mediated by the CDN infrastructure. The CDN operator owns a given number of surrogate servers, scattered all over the world, which thus perform content caching and replication, improving the Quality of Service (QoS) of the consumers by serving the requests of the clients in the neighbourhood[18] . One of the main features of CDNs is that they do not change the current key network protocols, but they rather offer countermeasures to address the peculiar characteristics of the Internet infrastructure that limit its effectiveness when performing content distribution.

.

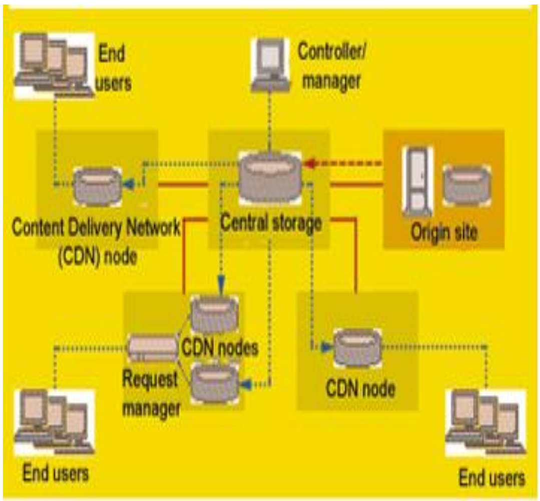

Fig:1.1 CDN Infrastructure

II. LITERATURE REVIEW

NikolaosLaoutaris, GeorgiosSmaragdakis, Konstantinos Oikonomou, IoannisStavrakakis, AzerBestavros developed a scheme in which an initial set of service facilities are allowed to migrate adaptively to the best number so as to best service the current demand .The scheme is based on developing distributed versions of the UKM problem (for the case in which the total number of facilities must remain fixed) and the UFL problem (when additional facilities can be acquired at a price or some of them be closed down). Both problems are combined under a common framework with the following characteristics: An existing facility gathers the topology of its immediate surrounding area, which is defined by an *r-ball* of neighbors – nodes that are within a radius of *r* hops from the facility. The facility also monitors the demand that it receives from the nodes that have it as closest facility. It keeps an exact representation of demand from within its *r*ball, and an approximate representation for all the nodes on the *ring* of its *r*-ball (nodes outside the *r*-ball that receive service from it). In the latter case, the demand of nodes on the *"skin"* of the *r*-ball is increased proportionally to accountfor the aggregate demand that flows in from outside the *r*-ball through that node. When multiple *r*-balls intersect, they join to form more complex *r-shapes*. The observed topology and demand information is then used to re-optimize the current location (and optionally the number of) facilities by solving the UKM (or the UFL) problem in the vicinity of the *r*-shape.

The availability of a set of network hosts upon which specific functionalities may be installed and instantiated on demand is envisioned.The term "Generic Service Host" (GSH) to refer to the software and hardware infrastructure necessary to host a service. The implementation of the above-sketched scenarios requires each GSH to be able to construct its surrounding AS level topology up to a radius *r*. This can be achieved through standard topology discovery protocols Also, it requires a client to be able to locate the facility closest to it, and it requires a GSH to be able to inform potential clients of the service regarding its W or SB status. Both of these could be achieved through standard resource discovery mechanisms like DNS re-directionor proximity-based anycast routing.

In this study authors described a distributed approach for the problem of placing service facilities in large-scale networks. The scalability limitations of classic centralized approaches are overcome by re-optimizing the locations and the number of facilities through local optimizations which are refined in several iterations. Re-optimizations are based on exact topological and demand information from nodes in the immediate vicinity of a facility, assisted by concise approximate representation of demand information from neighboring nodes in the wider domain of the facility. Using extensive synthetic and trace driven simulations we demonstrate that our distributed approach is able to scale by utilization limited local information without making serious performance sacrifices as compared to centralized optimal solutions. Konstantinos Oikonomou and Ioannis Stavrakakis In this paper, the problem of determining the optimal location of a service facility is revisited and addressed in a way that is both scalable and deals inherently with network dynamicity. In particular, service migration which enables service facilities to move between neighbour nodes towards more communication cost-effective positions, is based on local information. The migration policies proposed in this work are analytically shown to be capable of moving a service facility between neighbour nodes in a way that the cost of service provision is reduced and under certain conditions , the service facility reaches the optimal (cost minimizing) location, and locks in there as long as the environment does not change; as network conditions change, the migration process is automatically resumed, thus, naturally responding to network dynamicity under certain conditions.

Service migration was explored in this paper as a way of addressing the service placement problem in large scale and dynamic networking environments. Unlike classical approaches that require the knowledge of global network topology and detailed service demands, the proposed approach requires only local topology information (neighbors of the service hosting node) and aggregate service demands that become readily available to the service hosting node. Furthermore, as the network conditions change, one of the three migration policies presented here (Migration Policy S) inherently incorporates these changes in determining the migration path to follow in each step, as opposed to requiring to take a snapshot of the environment (network topology and service demands) at a certain time and to solve the corresponding large optimization problem. Michele Mangili, Fabio Martignon and Antonio Capone proposed a novel theoretical framework to analyze the performance of a Content-Centric Network and to provide clear comparative results with a Content-Distribution Network**.** This paper give clear answers to this critical issue by proposing a methodology to assess how the innovative design of Content-Centric Networking behaves as opposed to the solution proposed by Content-Distribution Networks.The authors developed a novel optimization model to study the performance bounds of a Content-Centric Network, by addressing the joint object placement and routing problem.Further introduced three comparative models that well describe 1) a Content-Distribution Network, 2) a traditional IP-based network, and 3) a Content-Centric Network whose caches are prepopulated with given contents.

Finally, discuss the numerical results showing the performance bounds of this revolutionary paradigm. The authors discovered that: 1) a Content-Centric Network with small caches can provide significant performance gains compared to a traditional IP-based network; 2) for large amounts of caching storage,the benefits of using sophisticated cache replacement policies are dramatically reduced and 3) in some scenarios, a Content- Distribution Network with few replica servers can perform better than a Content-Centric Network, even when the total amount of available caching storage is exactly the same.

Performance bounds were derived by addressing the joint object placement and routing problem. By performing an extensive numerical analysis we discovered that: 1) The presence of a distributed cache in both the CCN and CDN architectures can have significant benefits for the QoS of the network since it makes possible to accommodate much higher traffic demands even when few objects are stored in the nodes. 2) For large caching storage, the benefits of using sophisticated cache replacement policies are dramatically reduced. 3)A Content-Distribution Network can provide slightly better performance than a CCN, even when the total amount of caching storage deployed in the network is exactly the same. This is possible due to the fact that a CDN provides the additional degree of freedom to choose the location of the distributed cache.this is the first attempt to model and compare the performance of a CCN with respect to that of a CDN without using a simulated model.

III. PROBLEM DEFINITION

A major change in the emerging networking landscape concerns the role of end-user, who is no more only content consumer but also generator. A major challenge for this emerging paradigm is how to make these exploding in numbers, yet individually of vanishing demand, services available in a cost-effective manner; central to this task is the determination of the optimal service host location.

The optimal placement of up to k service instances is typically treated as a k-median problem. Input to the problem is the network topology and service demand distribution across the network users. The focus instead is on the 1-median problem variant that seeks to minimize the access cost of a single service replica since it matches better the forthcoming User Generated Service paradigm[1]. This will enable the generation of service facilities in various network locations from a versatile set of amateur user service providers. The huge majority will be lightweight services requiring minimum storage resources and addressing relatively few users in the ‘‘proximity’’ of the user service provider, either geographical or social, so that the replication across the network would not be justified.

Here this problem is formulated as a facility location problem and devise a distributed and highly scalable heuristic to solve it. Key to this approach is the introduction of a novel centrality metric. Wherever the service is generated, this metric helps to:(1)Identify a small sub graph of candidate service host nodes with high service demand concentration capacity(2)Project on them a reduced yet accurate view of the global demand distribution(3)Pave the service migration path towards the location that minimizes its aggregate access cost over the whole network.

In each service migration step, the metric serves two purposes. Firstly, it identifies those nodes that contribute most to the aggregate service access cost and pull the service strongly in their direction; namely, nodes holding a central position within the network topology and/or routing large service demand amounts. Secondly, it correctly projects the attraction forces these nodes exert to the service upon the current service location and facilitates a migration step towards the optimal location.

IV. PROPOSED SYSTEM

In autonomic approach, propose a scalable decentralized heuristic algorithm that iteratively moves services from their generation location to the network location that minimizes their access cost[5]. 1-median formulation is employed as more suitable for the user-centric service paradigm. Contrary to centralized approaches, where a super-entity with global information about network topology and service demand solves the problem in a single iteration, we let it migrate towards its optimal location in a few hops. In each iteration, a small-scale 1-median problem is solved so that the computational load is spread along the migration path nodes.

In proposed system the approach is to replace the one-shot placement of service with its few-step migration towards the optimal location. Thus, we end up solving locally a few 1-median problems of dramatically smaller scale and complexity compared to the global 1-median problem. For every transit location of the service in the network, a centrality metric : 1) identifies nodes in graph as candidates for hosting the service in the next iteration;2) simplifies the projection of the service demand from the rest of the nodes on these candidates.

Centrality-Driven Distributed Service Migration Algorithm (cDSMA) progressively steers the service towards its optimal location via a finite number of steps.

Step 1) Initialization.

The first algorithm iteration is executed at node s in G˜ that initially generates the service facility. In subsequent iterations, the new reference node is the one each time hosting the service.

Step 2) Metric computation and 1-median subgraph derivation.

Next, the wCBC(u; s)metric is computed for every node u in the network graph G˜.Nodes in G˜ featuring the top α% wCBC values, together with the node currently hosting the service host form the 1-median subgraph G˜Host over which the 1-median problem will be solved Clearly, its size and the algorithm complexity are directly affected by the parameter choice.

Step 3) Mapping the demand of the remaining nodes on the sub graph.

To account for the contribution of the ‘‘outside world’’ to the service provisioning cost, the demand for service from nodes in G /G˜ Host is mapped on the G˜Host ones. To do this correctly and with no redundancy, the algorithm credits the demand of some outside node z only to the first ‘‘entry’’ G˜Host node encountered on each shortest path from z towards the service host. Thus, the weights w(n) for calculating the service access cost at 1-median problem solution and service migration to the new host node.

Step 4) 1-median problem solution and service migration to the new host node.

Any centralized technique may be used to solve this small-scale optimization problem and determine the next best location of the service in G˜ Host. s is the current service location as long as node s 1) yields higher cost than the candidate Host node the candidate Host has not been used as a service host before the service is moved there and the algorithm iterates through steps 2-4..

Step5) Data Transfer

cDSMA steers the service to the lowest-cost location. The service will be accessed from the nearest neighbour having the service or the replica.

The entire work is divided into four modules.

1. Topology Generation

The topology is auto generated. The network topology is represented as an undirected connected graph G=(V,E). A subset Vs⊆ V of the total network nodes are enabled to act as service host sites and along with the set Es ⊆ E of edges linking them, form the generally disconnected, subgraph G˜=(Vs,Es). Each potential service host k ∈Vs may serve one or more users attached to some network node n ∈ V and accessing the service with different intensity, generating demand w(n) for it. The goal is to find the service host k that minimizes the aggregate, i.e. by all network users ,access cost of a service facility

$$Cost(k) = \sum w(n).d(k,n)$$

where d(k,n) is the distance.

The module addresses the issues like service request, node request, node creation, range of node, position of node and ip address of node.

Nodes having higher number of links is a key in establishing links between other nodes. Such nodes could also be probably be major players in the solution of one median problem[3]. The main objective is to identify them and use them to define a network topology yielding a small scale optimization problem.

2. Service Formation

Service migration solves the service placement problem in large scale and dynamic networking environment. The service is placed in some nodes that is created in the previous module[5]. The service node monitor the amount of data exchanged through its neighbour nodes associated with the particular service and decide on the service movement based on the information gathered through the monitoring process. The service is moved finally to the optimal service node and remains there as long as the network status do not change significantly. When changes occur the service moves towards the new optimal service node position.

An algorithm proposed ,centrality-driven Distributed Service Migration Algorithm (cDSMA) progressively steers the service towards its optimal location via a finite number of steps .The weighted Conditional Betweenness Centrality (wCBC) captures both topological and service demand information for each node. It is the centrality metric driving the migration process. The metric take into account the service demand that can be routed through the shortest path towards the service location.The metric can effectively identify directions of high demand attraction. It is used in each iteration to, first, select candidate service host nodes and, then, modulate the demand weights with which each one participates in the local 1-median problem formulation.

Input to the problem is the network topology and service demand distribution across the network users. The focus is on the 1-median problem variant that seeks to minimize the access cost of a single service replica since it matches better the forthcoming UGS paradigm. This will enable the generation of service facilities in various network locations from a versatile set of amateur user service providers. Their huge majority will be lightweight services requiring minimum storage resources and addressing relatively few users in the proximity of the user service provider, either geographical or social.

The module identifies the type of services, for example message service or uploading or downloading services. It should also take into account the number of services of each type.

3. Replica Allocation

Replication offers the potential to improve system scalability by distributing the load across multiple servers. In general, a client would experience shorter access latency if a replica of the requested object (e.g., a web page or an image) is placed in its closer proximity[19]. Therefore, the effectiveness of replication, to a large extent, depends on the locations where the replicas are placed. When there are k replicas of the content available the problem of their optimal placement is formulated by k median problem.

Here the object placement problem is addressed i.e., the replication entity is an object(service) replica. The object copy located at the origin server is called the *origin copy* and an object copy at any remaining server is called a *replica*.Here one median problem is addressed ,i.e. only one copy is generated. If there is an increase in demand for the service, then the replica placement should be done in such a way that the requests are served optimally. The replica can be accessed by all the requesting nodes from its nearest neighbour having the replica.

The entire process is explained with cDSMA algorithm.

Algorithm cDSMA in G˜(Vs,Es)

```
choose randomly node s
place SERVICE @ s
for all u ∑G˜do compute wCBC(u; s) ; set flag(u)=0
G˜s ← {α% of G˜ with top wCBC values}U {s}
for all u ∑ G˜s do
compute wmap(u; s)
weff (u; s) ← wmap(u; s) + w(u)
compute cost C(u) in G˜s
Host ← 1-median solution in G˜s
while CHost <Cs do
    if flag(Host)==1 then
    abort
else
    move SERVICE to Host; flag(s) =1
    for all u∑G˜ do compute wCBC(u; Host)
G˜Host ← {α% of G˜ with top wCBCvalues}U{Host}
    for all u∑G˜Host do
        compute wmap(u; Host)
        weff (u;Host) ← wmap(u; Host) + w(u)
        compute cost C(u) in G˜Host
            s ← Host
    Host 1-median solution in G˜Host
end if
end while
```

4. Demand Mapping

After the derivation of the 1-median subgraph, the current service host needs to further process the measurement reporting messages that correspond to the selected subgraph nodes. Besides being the basis for extracting the 1-median subgraph G˜Host in each algorithm iteration, the wCBC metric also eases the mapping of the demand that the rest of the network nodes in G\G˜Host induce on the 1- median subgraph.

This demand must be taken into account when solving the 1-median problem.This is done by modulating the original wCBC metric in accordance with two observations. Firstly, during the computation of the node wCBC values, the demand of a node z in G\G˜Host is taken into account in all the G˜ Host nodes that lie on the shortest path(s) of z towards the service host node t. Simply mapping the demand of z on all those nodes inline with the original wCBC metric, has two shortcomings: (a) when the demand of heavy-hitter nodes is distributed among multiple nodes, any strong direction(gradient) of heavy demand that would otherwise "pull" the service towards a certain direction, tends to fade out; (b) the cumulative demand that is mapped on all G˜ Host nodes ends up exceeding considerably the real demand a node poses for the service.

The services are granted according to the demand. This module display the output. The user can view the path through which the service being migrated and create a map according to the demand. During network migration, an automated map provides accurate visualization of the network before and after changes.

VI. PERFORMANCE ANALYSIS

Performance analysis involves gathering formal and informal data to help users and service providers define and achieve their goals. Performance analysis uncovers several perspectives on a problem or opportunity, determining any and all drivers towards or barriers to successful performance, and proposing a solution system based on what is discovered. The accomplishment of a given task measured against preset known standards of accuracy, completeness, cost, and speed. In a contrast, performance is deemed to be the fulfillment of an obligation, in a manner that releases the performer from all liabilities under the contract. The definition for performance analysis given is: A specific, performance based needs assessment technique that precedes any design or development activities by analyzing the performance problems of a work organization.

It distinguishes three basic steps in the performance analysis process: data collection, data transformation and data visualization. Data collection is the process by which data about program performance and obtained from an executing program. Data are normally collected in a file, either during or after execution, although in some situations it may be presented to the user in real time. Data transformations are applied, often with the goal of reducing total data volume. Transformations can be used to determine mean values or other higher order statistics or to extract profile and counter data from traces. Performance analysis is the provision of objective feedback to performers trying to get a positive change in performance. The analysis can either take place immediately following the performance i.e. side of the track, on the court, end of the pool, or can take place in the laboratory in a more controlled environment.

The location service is one of the major players for scalability. It has two different roles: at the location server domain, it must provide detailed local information and balance requests among local available servers; at a global level, each location server must provide a scalable global trading information service. The global load balancing is based on network proximity relations.

If the demand is very high, a server might become crowed with bounded clients. A redistribution of clients must then take place to speed up the client's service total response time. The server can unbind some of the clients, or can mutate itself. This will force a new resolution phase for all waiting clients and are distribution of the clients for the available servers.

Fig 1.2 shows the evolution over time of some averages during each measuring interval of: *New Clients*, the number of clients which entered the system; *Unbinds*, the number of clients unbound during the interval; *Pending Clients*, the number of clients waiting on queues (of both application and location servers); *Ending Clients*, the number of clients which died during the interval; and *Processing Capacity*, the maximum number of clients that the servers can process during the interval.

As soon as client requests start,the number of pending clients grows until a point in time where the processing power deployed is enough for the client load, *Tsetup*. After that it starts to decrease. TT continues to grow just for a short while after this point (the curves are almost equal because redistribution was used, as curve *Unbinds* show). With the reduction of the number of pending clients, the number of servers decreases. However, the processing capacity is always above the new client rate due to the 50% idle time allowed for each server (the minimum load threshold).

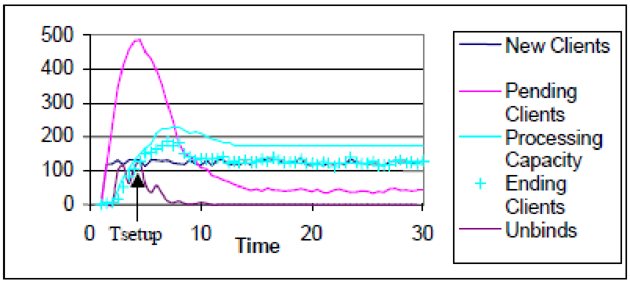

Fig:1.2

In figure 1.3 it compare the performance of existing and proposed system. The graph shows how the access rate is affected with respect to the number of host nodes in the subgraph. It is shown that as the size of subgraph increases the access rate also increase.ie, the service can be accessed with shorter delay as the size of subgraph increases.

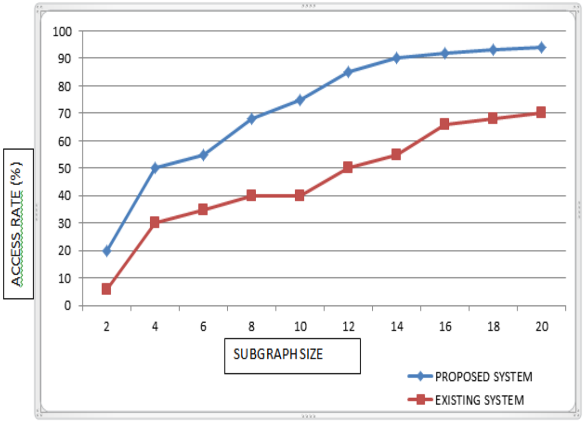

Fig:1.3

VII. CONCLUSION

Service deployment is a key problem in communication networks as it determines how efficiently the user service demands are supported. The performance of a network is increased by intelligently selecting which nodes are to host a particular service .A scalable and efficient approach was developed for optimal placement of service facilities and thus ensure minimal cost for accessing the service in dynamic network environments which enables the user to access the service with minimum cost. After the success of user-generated content, user-oriented service creation emerges as a new paradigm that will let individual users generate and make available services at minimum programming effort. Scalable distributed service migration mechanisms will be key to the successful proliferation of the paradigm.. Compared with distributed approaches that costrain the 1 median subgraph to the direct node neighbourhood, this mechanism yields better service placement.

ACKNOWLEDGEMENT

This work was supported in part by the Department of Computer Science & Engineering, SIST, Trivandrum. I would like to show my gratitude to Dr. C G Sukumaran Nair (HOD), Associate Professor, Sudha SK and Assistant Professor, Hamil Stanly, for their valuable guidance.

REFERENCES

[1] Panagiotis Pantazopoulos, Merkouris Karaliopoulos, Ioannis Stavrakakis “Distributed Placement of Autonomic Internet Services” IEEE transactions on parallel and distributed systems, Vol. 25, No. 7, July 2014

[2] C.A. La, P. Michiardi, C. Casetti, C. Chiasserini, and M. Fiore,‘Content Replication in Mobile Networks,’’ IEEE J. Sel. Areas Commun., vol. 30, no. 9, pp. 1762-1770, Oct. 2012.

[3] G. Smaragdakis, N. Laoutaris, K. Oikonomou, I. Stavrakakis, and A. Bestavros, ‘‘Distributed Server Migration for Scalable Internet Service Deployment,’’ IEEE/ACM Trans. Netw., doi: 10.1109/TNET.2013.2270440.

[4] P. Pantazopoulos, I. Stavrakakis, A. Passarella, and M. Conti,‘‘Efficient Social-Aware Content Placement for Opportunistic Networks,’’ in IFIP/IEEE WONS, Kranjska Gora, Slovenia, Feb. 3-5,2010, pp. 17-24.

[5]K. Oikonomou and I. Stavrakakis, ‘‘Scalable Service Migration in Autonomic Network Environments,’’ IEEE J. Sel. Areas Commun.,vol. 28, no. 1, pp. 84-94, Jan. 2010.

[6]P. Pantazopoulos, M. Karaliopoulos, and I. Stavrakakis,‘‘Centrality-Driven Scalable ServiceMigration,’’ in Proc. 23rd ITC, San Francisco,CA, USA, 2011, pp. 127-134.

[7]K. Oikonomou, I. Stavrakakis, and A. Xydias, “Scalable Service Migration in General Topologies,”

The Second International IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2008), Newport Beach, California, 23 June, 2008

[8] D. Trossen, M. Sarela, and K. Sollins, ‘‘Arguments for an Information-Centric Internetworking Architecture,’’ SIGCOMM Comput. Commun. Rev., vol. 40, no. 2, pp. 26-33, Apr. 2010

[9] M.E.J. Newman, ‘‘The Structure and Function of Complex Networks,’’ SIAM Rev., vol. 45, no. 2, pp. 167-256, 2003.

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

[11] . V. Jacobson, D.K. Smetters, J.D. Thornton,M. Plass,N.H. Briggs, and R.L. Braynard ‘‘Networking Named Content,’’ in Proc. 5thACM CoNEXT, Rome, Italy, Dec. 2009, pp. 1-12..

[12] T. Sproull and R. Chamberlain, ‘‘Distributed Algorithms for the Placement of Network Services,’’ in Proc. ICOMP, LasVegas,NV, USA, July 2010, pp. 1-8.

[13] S. Pandit and S. Pemmaraju, ‘‘Return of the Primal-Dual: Distributed Metric Facility Location,’’ in Proc. ACM PODC, 2009, pp. 180-189.

[14] G. Wittenburg and J. Schiller, “A survey of current directions in service placement in mobile ad-hoc networks,” in *IEEE PERCOM ’08*, Hong Kong, March 17-21 2008, pp. 548–553

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

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

[17] C. Boldrini, M. Conti, and A. Passarella, “Content Place: social aware data dissemination in opportunistic networks,” in Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems. ACM New York, NY, USA, 2008, pp. 203–210.

[18] G. Pallis and A. Vakali, “Insight and Perspectives for Content Delivery Networks,” Communications of the ACM, vol. 49, no. 1, pp. 101–106, 2006.

[19] J. Kangasharju, J. Roberts, and K. W. Ross, “Object Replication Strategies in Content Distribution Networks,” Computer Communications, vol. 25, no. 4, pp. 376–383, 2002.

[20] S. Borst, V. Gupta, and A. Walid, “Distributed Caching Algorithms for Content Distribution Networks,” in Proc. of the 29th IEEE Conference on Information Communications (INFOCOM’10), pp. 1478-1486, San Diego, California, USA, 2012.