Volume 2, Issue 9, September 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A scalable peer-to-peer lookup protocol for Internet applications

Gade Krishna[1] M.Tech.(CSE) VCOE,KNR,AP,INDIA
V.Kishore[2] Assoc.Professor VCOE,KNR,AP,INDIA
S.Naveen Kumar[3] Assoc. Professor SCITS,KNR,AP
O.Srinivas[4] M.Tech.(CSE) SCCE,KNR,AP
B.Praveen[5] Asst. Professor SCITS,KNR,AP

Abstract: **while peer-to-peer networks are mainly used to locate unique resources across the Internet, new interesting deployment scenarios are emerging. Particularly, some applications (e.g., VoIP) are proposing the creation of overlays for the localization of services based on equivalent servants (e.g., voice relays). This paper explores the possible overlay architectures that can be adopted to provide such services, showing how an unstructured solution based on a scale-free overlay topology is an effective option to deploy in this context. Consequently, we propose EQUATOR (Equivalent Servant Locator), an unstructured overlay implementing the above mentioned operating principles, based on an overlay construction algorithm that well approximates an ideal scale-free construction model. We present both analytical and simulation results which support our overlay topology selection and validate the proposed architecture.**

Keywords: **Distributed services, equivalent servants, resource sharing, peer-to- peer overlays, scale-free topology**

I. INTRODUCTION

While in the past few years the resource sharing services across the Internet focused on generic storage, content CPU cycles, the recent emerging of the cloud computing paradigm [1][2] might push this vision even further. According to this scenario, the world will be populated by thin and light computing devices acting mainly as frontends, while the computation and the user's data reside elsewhere, in the "cloud". In those services, two groups of entities are defined: "users", that ask for a given service, and "servants" that are actually in charge of providing the service. Servants can be composed of millions of processing platforms either sparse across the Internet, or concentrated in special datacenters. Users do not care about their physical location: they are interested in getting the service, no matter which servant is actually providing it.

This paper focuses on services provided by equivalent servants [1][4] and models and analyzes the performance of structured and unstructured [2] overlays when used to provide such services. We demonstrate that the architecture chosen for the P2P network has a huge impact on the overall performance of the service..

II. RELATED WORK

We propose a scheme for CPU cycle sharing over an unstructured P2P network [2]. We consider the unbalanced node degree distribution, which may result in real overlay networks, as a possible obstacle to the lookup effectiveness of the system and, consequently, they propose mechanisms to overcome these limitations. In this paper, we show instead how an unbalanced node degree distribution (specifically, a scale-free topology), if properly exploited, ensures high lookup performance. Peer-to-peer SIP proposes [3] to use a DHT to support lookups of relay nodes among all the equivalent participating peers, which can be done by randomly selecting a target node and then moving over the DHT to reach this target. This paper focuses on services based on equivalent servants and brings several contributions to the existing work on this topic. First, we compare the possible overlay architectures to support our class of services and we show, through extensive analytical and simulation studies, that an unstructured overlay based on a scale-free topology is an interesting solution in this context. Furthermore, we show the corresponding penalty in case a DHT architecture[3][9] is chosen, we propose a novel overlay construction algorithm[9][10] which is (i) suitable for implementation in real networks, (ii) supports a generic service, and (iii) approximates an ideal well-known scale-free construction model. Third, we analyze different network scenarios by varying the servant characteristics [10](e.g., their lifetime), which provides an insight of the possible performance of different services in our context.

III.OVERLAY ARCHITECTURE OPTIONS

Since the underlying overlay architecture has a huge impact on the performance of the offered service and on the features that can be guaranteed to the users, this section compares the structured and unstructured approaches with respect to their capability to support services based on equivalent servants. In particular, we focus on the service lookup performance offered by different architectures, presenting some analytical and simulation [3][4]results which demonstrate that an unstructured overlay based on a scale-free topology is a good choice for handling our service.

A. Structured overlays

We first investigate the possibility to deploy a structured overlay [3] based on a general DHT, as it has been proposed in for the P2PSIP architecture. Since in our scenario all peers provide the same functionality the number of copies predominates over the number of distinct services and therefore the ability of DHTs to locate a specific resource is of little help Therefore, proposes to use the DHT in a more clever way: queries are performed by randomly selecting a target key and then moving in the overlay to reach this target. Since it does not cause further complexity and possibly improves the system performance, we introduce an additional feature to this querying mechanism: during the lookup process [2][3], any node encountered along the path is checked for availability and can be selected as a servant for the querying user. Notice that this operating mode makes the approach independent of the adopted DHT. In fact, only the overlay topology is of interest in our context. In other words, we adopt the topology of a generic DHT, with a fixed number of neighbors for each node, but we use a different routing mechanism. This solution will be however referred to as DHT in the rest of the paper.

The idea of using a DHT for our scenario of equivalent servants is especially interesting in case a DHT has to be implemented anyway for some other services.

B.Unstructured overlays

An efficient unstructured overlay [5][6][7] is characterized by high lookup performance and small amount of traffic required to maintain the overlay. Both parameters are influenced by the topology and the operating principles of the overlay. This section elaborates on these aspects in the context of services based on equivalent servants, proposing to adopt a scale-free topology and motivating this choice.

An interesting lookup solution that avoids the deleterious traffic overhead generated by flooding-based queries is the adoption of a service lookup based on *random walks* encompassing a bounded number of nodes. Within this technique, the service request is forwarded, at each node, to a peer randomly selected among its neighbors. If the encountered node is available or knows an available servant, the procedure terminates. The knowledge of nodes can be improved through proper advertisement messages containing the node itself and/or other participating peers, thus implementing a so called *epidemic dissemination algorithm [5]*. The effectiveness of random walks depends on the overlay topology adopted in the system. Among other possibilities, a scale-free topology may offer interesting features. In a scale free network, the node degree distribution follows a power-law $(n) = cn^{-\gamma}$, where (n) is the probability that a node has n connections and c is a normalization factor. Hence, only few nodes (usually referred to as *hubs*) have a high degree, i.e., are aware of the existence of a large number of participating peers. The idea is that directing random walks toward hubs means looking for the service where there is a great knowledge of servants. This ensures high lookup performance with respect to an overlay based on a balanced degree distribution where service requests are randomly distributed among peers. This result derives from a well-known property of queuing systems, which says that a unique M/G/k/k queuing system servicing an arrival process with rate λ performs better than k separated M/G/1/1 systems each one servicing an arrival process with rate λ/k. In essence, concentrating the traffic on some nodes that have a deep knowledge of the network provides better performance than accurately distributing the requests among all nodes, as random solutions try to do. This extends the results obtained by Adamic in the context of traditional file lookups in P2P systems, which demonstrated the effectiveness of random walks in scale-free networks [10] due to the greater knowledge of resources available at the hubs. In order to achieve high lookup performance, hubs should have a deep knowledge about the other participating peers: the greater the number of peers known by a given node, the higher the probability for a user to find an available servant in a short time. Since the epidemic dissemination is based on flooding [5], the overlay topology has a deep impact not only on peers known by each node, but also on the resulting network efficiency. In fact, the greater the average path length between nodes, the higher the depth of the flooding that is needed for an adequate spread of the information, which may cause an unsustainable load on the network. The scale-free topology also ensures a good efficiency of epidemic dissemination algorithms as exhibits a small average path length.

One of the most popular mechanisms to build a scale-free network was proposed by Barabási and Albert and for this reason is referred to as Barabási-Albert model. Let m denote the out-degree of a node and d denote its in-degree. The Barabási-Albert model requires a set of m_0 nodes to be already in the system at the beginning of the process. Then, each entering node connects to m existing nodes, chosen proportionally to their popularity. This process is known as *preferential attachment*. This network formation algorithm results in a scale free network[9] characterized by a node degree distribution $p(n) = cn^{-3}$ and an average path length which behaves as $\frac{\ln N}{\ln \ln N}$.

C.A lookup performance model

This section compares the above architectures with respect to their capability in locating an available servant. This result is achieved by defining a simple analytical model that derives the average blocking probability (i.e., the probability for a service request to fail because no available servant is found) achieved by each architecture.

1) Model overview: From our point of view, the length of the path that a service request has to follow to reach a target key in a DHT and the depth of a random walk over an unstructured network have a similar meaning: they represent the amount of hops that a service request can encompass without success before the request has to be considered blocked. Hence, without losing in generality, we denote these two overlay parameters by a common variable, namely D_l, generally defined as the maximum depth of a service lookup. D_l is a fixed value in an unstructured network, while is variable and

$O(\log N)$ in a DHT. For the sake of simplicity, we consider only the case $D_l = 1$ in this model. Within the unstructured approach, we also assume a dissemination depth (i.e., the time-to-live of advertisement messages, denoted as T_d) not greater than 2 hops, as larger values would result in an excessive dissemination traffic overhead in the network. This assumption is confirmed by the guidelines of the Gnutella protocol [8], in which the depth of the dissemination algorithm is set to a maximum of 2 hops.

Let $V = \{v_1, v_2, \ldots,\}$ denote the set of participating peers offering the service, and S_i denote the set of servants indexed by a given node v_i (including the node itself), i.e., the set of peers that the node v_i can offer to a querying user in the tentative of satisfying her service request. The idea is that, whenever a service request reaches a node v_i of the overlay, such request is satisfied if a servant $s_i \in S_i$ is available.

Hence, under the assumption $D_l = 1$ and if service requests are supposed to arrive at nodes according to a Poisson process, each node can be modeled as an M/G/k/k queuing system, i.e., a buffer-less system offering k equivalent servers, as also briefly described in Section III-B. End-systems may be part of the overlay if the offered service consumes a small fraction of the available resources, so that local users are not penalized. Hence, we suppose that a node can be a servant only for one user at a time, i.e., for a given node v_i, $k_i = |S_i|$. This could not be the case in some scenarios, where the offered service consumes a very low percentage of resources. However, it is worth noticing that our model is still valid: if each node can support n service instances, we would have $k_i = n\,|S_i|$. Similarly, the analysis could be extended to the case where each node can handle a different number of service requests. However, this would complicate the analysis without adding any significant contribution to the comparison among the architectures considered in the paper. In an M/G/k/k queuing system, the probability that a service request fails (i.e., the blocking probability, which we denote as P_b) can be evaluated by using the well-known Erlang B formula. Let λ_i and μ_i denote the request arrival rate and the service rate at node v_i, respectively. For each node v_i we have

$$P_{bi} = \frac{\rho_i^{\,ki}/ki!}{\sum_{n=0}^{ki} P^{n/n!}} \qquad (1)$$

Where $\rho_i = \lambda_i/\mu_i$ is defined as the *service request load* at node v_i. Clearly, for a given node v_i, P_{bi} depends on ρ_i and on the amount of servants the node can offer, k_i. In the next sections we will derive ρ_i and k_i for both the structured and the unstructured scale-free approach. This will be used to calculate the average blocking probability of the system, which allows us to quantitatively compare the two approaches under examination when used to locate equivalent servants. In particular, if ρ_T is the total service request load offered to the overlay, the average blocking probability can be evaluated as

$$P_b = \sum_{i=1}^{N} \frac{\rho_i}{\rho_T} P_{bi} \qquad (2)$$

***2) Structured overlay model**:* From our point of view, a DHT can be modeled as a regular topology [8][9] where nodes have a fixed number of neighbors (the out-degree m) given by the size of the tables they use to route queries in the overlay. According to the servant lookup procedure presented in Section III-A, each encountered node along the path toward the target key can satisfy the service request only if the node is available, i.e., $S_i = \{v_i\}$ and, consequently, $k_i = 1$, $\forall v_i \in V$. We assume that incoming queries can enter the network at nodes selected randomly, as it may happen in real DHTs. Also remembering our main assumption $D_l = 1$, we have $\rho_i = \rho_T/N$, $\forall v_i \in V$, where N is the overlay size, i.e., the number of peers participating to the overlay. The average blocking probability is obtained by substituting k_i and ρ_i in (1) and (2).

3) Unstructured scale-free overlay model: In an unstructured scale-free overlay[9], the number of servants $|S_i|$ offered at a node v_i strictly depends on the amount of other peers which advertise themselves to the node, which varies according to the dissemination depth T_d adopted in the network. We consider a scale-free network constructed according to the Barabási-Albert model, which represents the scale-free construction algorithm we adopt as a reference in this paper; we consider $m_0 = m$ for simplicity.

Let A_i denote the amount of messages arriving at node v_i in one advertisement round from peers directly connected to node v_i and from which v_i is reached by a 1-hop-depth dissemination. In this simple case, a node v_i can receive advertisement messages only from its direct neighbors. Clearly, $A_i = d_i$, where d_i is the in-degree of node v_i. In the Barabási-Albert model, the in-degree of a node may vary whenever a new node joins the network. In particular, the probability $P_{n,}$ that an entering node v_n connects to an existing node v_i, thus modifying its degree, is given by

$$P_{n,} = \frac{d_i(n) + m}{2n}, \quad n > i, \qquad (3)$$

where $d_i(n)$ is the in-degree of node v_i when node v_n joins the network. This time dependence can be calculated by applying the *continuum theory*, introduced in for this purpose. The outcome of this theory is that, at a given time t, (t)

$= m\sqrt{(t/i)} - m$. Thereby, we can argue that, for a network size N (i.e., at "time N"), the amount of messages arriving at node v_i in one advertisement round when $T_d = 1$ is

$$A_i = m\sqrt{\frac{N}{i}} - \mathbf{m} \quad \text{------------------ (4)}$$

Analogously, let A'_i denote the number of messages arriving at node v_i in one advertisement round from peers connected to the direct neighbors of node v_i and from which v_i is reached by a 2-hop-depth dissemination. We define these nodes as "second-hop neighbors" of v_i. In this case, the calculation of the number of advertisement messages A'_i is more difficult as it is no longer deterministically related to the in-degree d_i of the node. For this reason, we focus our analysis on the average number of received advertisement messages, which is more tractable and does not preclude the validity of the model. In particular, considering that the average in-degree of the neighbors of node v_i can be evaluated as $\Sigma^N_{n=i+1}\, c\, d_n P_{n,i}$ and that, from the continuum theory, $P_{n,i} = (m/2)(ni)^{-0.5}$, we can derive the average number of advertisement messages generated by the second-hop neighbors of v_i as follows:

$$<A'_i> = d_i \sum^N_{n=i+1} cd_n P_{n,I} \approx d_I \int_{i+1}^{N} cd_n P_{n,I}\, d_n$$

$$\approx d_i\, cm^2\sqrt{(N/i)}\ \ [\ln(N/i) + \sqrt{(i/N)}\ -1 \quad \text{-----(5)}$$

adopted as an estimation of A'_i in the following. The normalization factor c can be evaluated by imposing $\Sigma^N_{n=i+1}\, cP_{n,} = 1$.

The set S_i of the servants indexed at a node v_i is composed of the node itself and the peers it discovers through the epidemic dissemination mechanisms. The m peers a node is connected to (i.e., its out-degree) are assumed to index other servants and contacted in the case the service cannot be satisfied at node v_i. It is worth noticing that, while for a node v_i, $|S_i| = A_i + 1$ if $T_d = 1$ (i.e., the number of servants indexed at the node is equal to the number of advertisement messages received in each advertisement round, plus the node itself) a similar consideration is in general not correct if $T_d = 2$, i.e., $|S_i| \neq A_i + A'_i + 1$. This is due to the fact that, when $m \geq 1$, the second-hop neighbors discovered may be not unique and we may count the same node twice. This may happen when two first-hop neighbors have an additional direct connection between them (therefore they both appear as second hop neighbors as well) or when the same second-hop neighbor is reached through two different first-hop neighbors.

Concerning the first type of duplicated node, we can obtain an approximate evaluation of the average number of links among the direct neighbors of a node v_i as follows:

$$\mathbf{L_i \approx 1/2} \int_{i+1}^{N} dl \int_{i+1}^{N} dn\, P_{li}\, P_{n,i} P_{l,n} \approx \frac{\boldsymbol{m^3}}{\mathbf{16}\boldsymbol{i}} \left[\boldsymbol{ln}\left(\frac{\boldsymbol{N}}{\boldsymbol{i}}\right)\right]^2,$$

derived from the definition of "average clustering coefficient" introduced in by considering that in the Barabási-Albert model a node v_n can be connected to node v_i only if $n > i$. Analogously, we can evaluate the average number of duplications involving a second-hop neighbor and two direct neighbors of a node v_i as follows:

$$\mathbf{L'_I} \approx \int_{i+1}^{N} dl \int_{i+1}^{N} \boldsymbol{dnP_{li}\, P_{(i+1),I}\, P_{n,l}\, P_{n,(l+!)}}$$
$$\approx m^4/16i^2\, [\ln(\tfrac{N}{i}) + \tfrac{i}{N}\ -1]$$

These parameters indicate, for each node, the average number of duplications resulting in the neighborhood of the node, from which an estimation of the number of non-unique discovered nodes could be derived. In particular, we have

$$k_i = a_i A_i + b_i\, \mathbf{(A'_i\ - L_i - L'_i\,) + 1}$$

where $a_i, b_i \in \{0, 1\}$, $a_i = 1$ iff $T_d \geq 1$, $b_i = 1$ iff $T_d \geq 2$. As mentioned before, we believe that a scale-free topology is especially advantageous if we direct service requests to hubs. Hence, we assume

$$\rho_i = \frac{k_i}{\sum_{n=1}^{N} k_n}\, \rho_T \quad \text{-------- (6)}$$

which corresponds to distributing incoming service requests among nodes proportionally to their popularity. The above derived values for k_i and ρ_i can be used to calculate the average blocking probability achieved in an unstructured scale-free overlay by applying (1)1 and (2). However, it is worth noticing that, for a given node v_i, if $v_i \in$ and $v_i \in$, v_i is a direct neighbor or a second-hop neighbor of v_j . This makes servants shared among several nodes and then introduces correlations which are not considered in the M/G/k/k system and in the Erlang B formula, which is not valid in such situations. We can model these correlations by introducing an additional load at all nodes, therefore taking into account the service requests directed to the nodes that share some servants. In particular, given an average service request load ρ_i

on a node v_i, the average contribution to the load on a node $v_j \in S_i$ due to v_i is. $^i_j p = (\frac{p}{k})_l$ .From (6), we can derive that this contribution is constant and equal to

$$\frac{P_T}{\sum_{n=0}^{ki} k_n}, \forall v_j \in V.$$

This considered, we can argue that the additional load to consider at a node v_i is $\rho_{iadd} = (m^2+m)$, where ρ_i is derived from (6). This approach for considering correlations deriving from the sharing of servants among nodes is approximated and, as shown in the next section, holds only for small values of m. Specific analytical work would be required to exactly model this phenomenon, which is however left for future work.

D. Lookup performance comparison

The above presented analytical model is used for comparing the structured [4][5]and the unstructured scale-free approaches concerning the average blocking probability they achieve. A network size of $N = 5000$ nodes is considered for this comparison. Furthermore, we assume an exponential service time distribution with rate $\mu_n = \mu, \forall : 1 \le n \le N$. In absence of any more detailed information about the possible service time distribution in this particular distributed service scenario, we consider this assumption a good approximation of the actual behavior of possible users, which could be involved in relatively short multimedia communications with higher probability, but also in longer sessions of video-streaming and on-line gaming. In essence, we customize our queuing system to an M/M/k/k. Notice how this does not influence our performance evaluation as the Erlang B formula is insensitive to the service time distribution. Fig. 1 compares the structured approach and the unstructured scale-free solution adopting an epidemic dissemination depth $T_d = 2$. In particular, it plots the average 1-hop blocking probability (i.e., the average blocking probability obtained when $D_l = 1$) achieved by these two types of network. Some values of the out-degree m are considered for the unstructured approach2. The figure shows how the unstructured scale-free approach based on epidemic dissemination and random walks significantly outperforms the modified DHT-based lookup over structured overlays introduced in Section III-A. To validate our model, we developed a custom, event-driven simulator implementing both the structured and the unstructured scale-free approach[9]. The former exploits a regular topology where service request load is randomly distributed among peers. Concerning the unstructured scale-free approach, the network topology is constructed according to the Barabási- Albert model and the service request load is distributed among peers proportionally to their in-

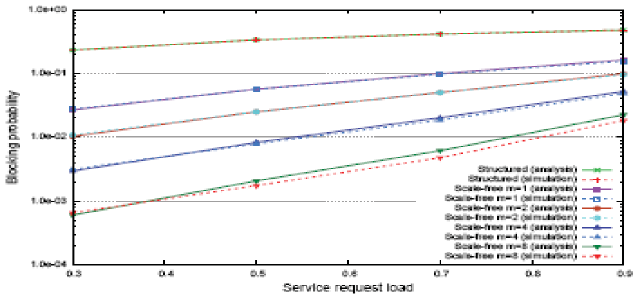

Fig.1.

Average 1-hop blocking probability: comparison between structured and unstructured scale-free overlays

degree, as specified by the model itself. Fig. 1 also compares our analytical model with the results obtained by simulation. We can observe how our approximated approach to address the correlations emerging from the presence of servants indexed at more than one peer,consisting in the introduction of additional load at nodes, is valid only for small values of m. For example, at a service request load $\rho_T = 0.7$, the analytical model provides an average blocking probability which is 20% higher than the real value derived by simulation when $m = 8$. However, it is worth noticing how small increments in the value of m result in sensible improvements of the lookup performance.

This is of great importance in our unstructured context, as the traffic overhead generated by flooding of advertisement messages is directly proportional to the out-degree m of the participating peers. Hence, we can concentrate on small values of m, which guarantee small traffic overhead together with excellent lookup performance. For this reason, we consider $m = 2$ in the following. To further extend our overlay comparison, we consider larger values of D_l. This was not included in our model because of the complex correlations that rise when service requests may experience more than one hop. In fact, requests may arrive at a node after being refused at previous hops, making an analytical modeling difficult. Consequently, we derive this result only by simulation[2]. Moreover, we include additional comparisons which may be of interest for our work. In fact, so far we considered the utilization of the DHT as proposed in However, a possible more effective approach may be to include epidemic dissemination in the structured overlay, so that nodes may increase the number of servants they can offer to querying users. Since in our context we are interested in the number of node edges (proportional to the number of servants discovered), rather than in the specific peers to which they point, such an approach is expected to have similar performance to an unstructured overlay implementing a *random graph* (also considered in this comparison for the sake of completeness). In fact, a random graph is by definition a quasi-regular topology where node degree assumes values close to the average degree m with high probability. Notice that this analogy does not hold in traditional file-sharing systems, where efficient lookups over structured overlays are guaranteed only if peers establish connections according to well-defined rules.

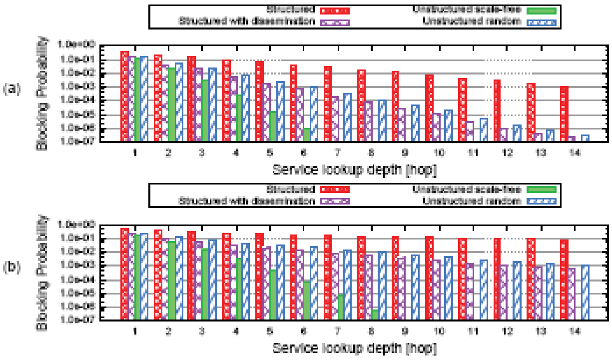

Fig.2

Average blocking probability as a function of the service lookup depth D_l: comparison among structured, structured with epidemic dissemination, unstructured scale-free, and unstructured random overlays; (a) $\rho_T = 0.6$; (b) $\rho_T = 0.9$.

Fig. 2 compares all these approaches concerning the average blocking probability achieved at different values of D_l in the two different service request load conditions $\rho_T = 0.6$ and $\rho_T = 0.9$. Besides confirming that the unstructured random solution and the structured approach enriched with epidemic dissemination perform similarly, the figure shows how the unstructured scale-free overlay outperforms other solutions. In particular, Fig. 2(b) shows that the lookup performance of a random walk does not degrade too much with the increasing of the traffic intensity, thus being able to effectively support also services requiring real-time lookups. In summary, the outcome of these analytical and simulation results is that, besides avoiding the complexity due to the maintenance of a structured network, an unstructured overlay based on epidemic dissemination and resulting in a scale-free topology is also preferable for the offered lookup performance.

In particular, we observed how such a system can guarantee good lookup performance even in presence of small values of m and T_d ($T_d \leq 2$ in these examples), which are instrumental to limit the overhead in the network due to the flooding of advertisement messages. As described in Section III-B, these properties derive from the large availability of servants at the hubs and the small diameter of scale-free networks.

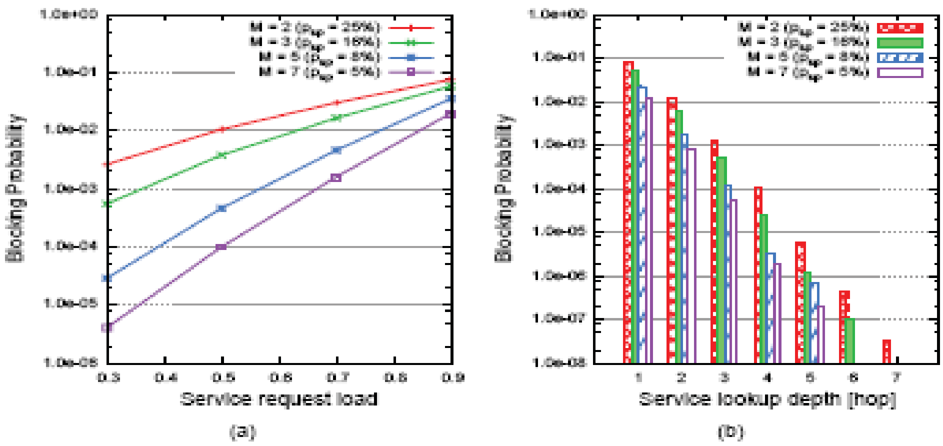

Fig. 3

Lookup performance of a hierarchical scale-free overlay: (a) Average 1-hop blocking probability; (b) Average blocking probability as a function of the service lookup depth D_l ($\rho_T = 0.6$).

E. Further properties of the unstructured scale-free overlay

Scale-free networks[8][]9offer some other properties which could be interesting for the deployment of service-oriented overlays.First of all, we show how the average blocking probability can be further lowered by reducing the number of nodes from which users can start random walks in order to locate a servant. In particular, since one of the key ideas under the adoption of a scale-free topology is to preferably direct queries toward hubs, we can force users to direct their service requests to nodes whose in-degree is greater than a given value M. In essence, service requests can be directed to a percentage $p_{sp} = (m^2/(M + m)^2) * 100$ of nodes, as can be derived from the continuum theory. Fig. 3(a) and Fig. 3(b) plot the average 1-hop blocking probability and the average blocking probability as a function of the random walk depth D_l, respectively, achieved by the unstructured scale-free network for some values of M when $m = 2$. A service request load $\rho_T = 0.6$ is considered in Fig. 3(b). We can observe how small values of M, resulting in percentages p_{sp} not lower than 5%, significantly improve the lookup performance of the scale free overlay. These are admissible values if we consider that very few nodes handle lookup requests in existing hierarchical file-sharing systems. Moreover, as it will be clearer in the following, handling a service requests[10] means processing a short packet and replying with some peer descriptors (i.e., a few hundreds of bytes)..

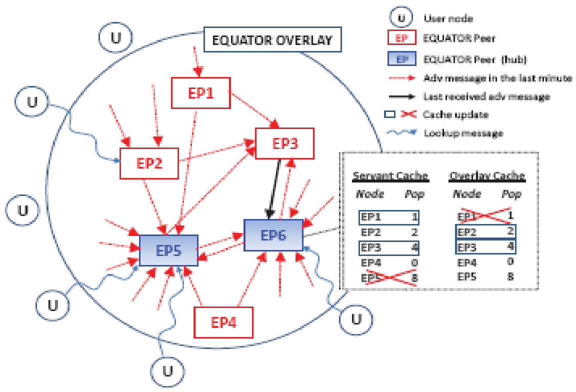

Fig. 4
EQUATOR architecture.

IV. EQUATOR

The previous section demonstrated the effectiveness of an unstructured network based on a scale-free topology[8][9]. However, both the Barabási-Albert model, adopted for the scale-free construction, and the lookup mechanisms deriving from this approach make some assumptions that cannot be satisfied in the real world; particularly, we envision four problems in the model that require some real-world adaptations. In fact, the Barabási-Albert model requires *(i)* a global knowledge of nodes and *(ii)* their popularity in order to perform the preferential attachment; *(iii)* hubs are supposed to index an arbitrarily large number of servants, which are used to satisfy incoming service requests; finally, *(iv)* nodes are considered static, i.e., the model does not consider nodes joining and leaving the network..

A.EQUATOR Bootstrap Service

In real P2P networks, entering nodes cannot have any knowledge about the existing overlay and therefore a Bootstrap Service is required in order to give such nodes the opportunity to join the network.

In EQUATOR, we prefer a more flexible approach that relies on multiple bootstrap servers reachable through appropriate DNS records (e.g., `SRV` entries), thus guaranteeing redundancy and load balancing.

B.Node popularity

In a network based on epidemic dissemination, nodes send advertisement messages to other nodes in order to maintain the overlay. Although the details of this advertisement process will be presented in Section IV-C, we need to define first a feasible method for computing the popularity of nodes, which is one of the crucial points of the Barabási-Albert model because it is at the foundation of the preferential attachment policy and hence of the scale-free construction mechanism. In a scale-free topology the popularity is equivalent to the in-degree of the node. Since it is unfeasible for an EQUATOR node to be aware of its in-degree, EQUATOR adopts as popularity metric the number of advertisement messages a node receives, which is proportional to its in-degree.

C.Overlay knowledge and advertisement

Each node in the overlay maintains two different node caches: a *servant cache* and an *overlay cache*. The former contains the set S_i of servants indexed by a peer v_i and it is populated by nodes that are lightly loaded with high probability, i.e., nodes (often leaves) that are most appropriate for satisfying an incoming service request. The latter contains a subset of the participating peers representing the entire overlay, among which the node selects the m peers to connect to. Hence, it includes nodes of different popularity in order to better represent the overlay. We denote by τ_{sc} the size of the servant cache and by τ_{oc} the size of the overlay cache. At each advertisement round (which we suppose to occur every t_{adv} minutes), an EQUATOR node sends an advertisement message (i.e., it "connects") to m peers in its overlay cache, chosen with a probability proportional to their popularity and hence according to the preferential attachment mechanism. These messages contain a list of tuple `<node, popularity, ttl>`: n_{sc} entries are selected as the less popular peers present in the servant cache, while n_{oc} entries are randomly selected from the overlay cache. This is done to give nodes the opportunity to learn both servants that are available with high probability (i.e., the leaves) and a set of nodes of different popularity to improve their local representation of the overlay. In fact, nodes that receive the message insert the n_{sc} entries in the servant cache and the n_{oc} entries in the overlay cache. When caches are full, the n_{sc} entries replace the most popular peers of the servant cache, while the entries replaced by the new n_{oc} nodes in the overlay cache are chosen randomly. Notice that the removal of oldest entries (as proposed in CYCLON [27]) is not a good policy in EQUATOR as it is necessary to maintain the above popularity distributions in the caches. However, entries expire after `ttl` seconds in order to purge old nodes from the cache (if not refreshed) and avoid zombies. When the dissemination depth $T_d > 1$, nodes along the dissemination path also insert themselves in the advertisement messages before forwarding the message to the next hop.

In EQUATOR, the knowledge of the network at any time t is limited to a few nodes, i.e., the ones that are in the two caches. Apparently, this is a radical departure from the scale free model in which nodes have the knowledge of the entire network. However, the advertisement policies implemented in EQUATOR allows a frequent update of the two caches, therefore changing the known peers over time. In fact, each node periodically advertises itself and some peers contained in its two caches, so that peers receiving advertisement messages can update their knowledge of the network by filling up, and possibly refreshing, their caches. Refresh is the key technique that allows the deployment of small caches, which limits overheads due to both cache management and advertisement and lookup traffic (all nodes in the servant cache have to be contacted during the lookup procedure, as described in Section IV-E). Furthermore, it reduces the possibility to have an old servant, which may be dead or currently unavailable (actually servicing a request), in the servant cache. In fact, a frequent cache refresh ensures the set of indexed servants changes frequently, resulting in a sort of round robin among them. Since the cache refresh rate at a node is proportional to the number of advertisement messages received and, consequently, to its popularity, this effect is maximized at the hubs, which have the opportunity to virtually offer a large number of servants, notwithstanding the limited size of the servant cache

E.Service lookup procedures from normal users

While the overlay contains all the peers that are available to offer some of their resources (i.e., are potential servants), many hosts may join the system as normal users in order to simply exploit the overlay services and without taking an active part in the overlay.

Users are most interested in service lookup functionalities and therefore have an advantage at connecting to peers that know many servants. In fact, in our model service requests are distributed among the participating peers proportionally to their popularity, i.e., requests are preferably directed to hubs. Consequently, preferential attachment is beneficial also for users and therefore we need to implement an approximation of this algorithm also with respect to these nodes. The service lookup procedure we defined for normal users works as follows. Each user maintains a node cache, referred to as *lookup cache*. Whenever a user logins in EQUATOR, her EQUATOR instance connects to the Bootstrap Service and retrieves the initial m_0 nodes. The user node selects one of them randomly and downloads its overlay cache. This procedure is repeated periodically in order to guarantee both the user node to have up-to-date knowledge of existing peers and service lookups to be well distributed among the peers.

V. EQUATOR Simulation Results

This section presents some simulation results on the EQUATOR architecture. We first validate our overlay construction algorithm, which we show to result in a scale-free topology. We also show how EQUATOR [9][10] is comparable to the ideal Barabási-Albert network in terms of lookup performance. We then elaborate on the system parameters, also focusing on the lookup and advertisement overhead at nodes. Finally, we investigate the behavior of our solution in different scenarios triggered by different kinds of peers.

A. Simulation background

To perform our simulations, we developed a custom, event driven simulator implementing the EQUATOR algorithms presented in the previous section. The simulator considers two types of nodes: participating peers and user nodes. The former are part of the EQUATOR overlay, while the latter represent the customers that need to exploit the offered service. Participating peer arrivals are modeled using a Poisson process, while we consider several distributions for peer lifetimes in order to investigate the behavior of EQUATOR in different scenarios. User node arrivals are modeled using a Poisson process, while user node lifetimes are assumed to be exponentially distributed. Once entered the network, user nodes run the lookup cache population algorithm presented in Section IV-E.

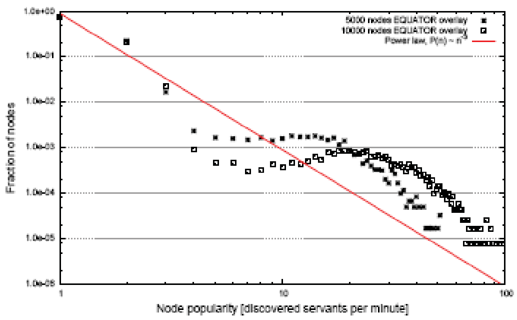

Fig. 5.
Node popularity distribution.

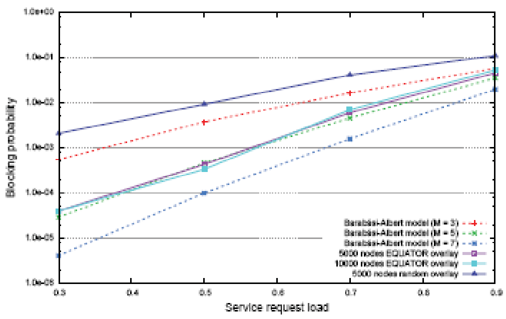

Fig. 6. 1-hop average blocking probability

We model service requests with a further Poisson process. Whenever a service request is scheduled, it is associated with one of the user nodes currently present in the network, which immediately starts a lookup procedure. To be compliant with the assumptions introduced in Section III-C, the service duration is exponentially distributed. We consider several service request rates, ranging from 50 to 150 requests/min. These values result in a service request load $\rho_T = 0.3 \div 0.9$. A single Bootstrap Server is adopted for simplicity. Incoming nodes, they are participating peers or users, contact this server and retrieve the m_0 registered peers. Different values for the overlay size N are considered, obtained by adopting a proper average peer arrival rate which, coupled with the average peer lifetime, results in an overlay of about N peers in the steady-state. Concerning the other system parameters, we set $\tau_{sc} = \tau_{oc} = 20$ nodes and $t_{adv} = 30$ min, which Section V-D will show to be proper values for the EQUATOR overlay.

B.Overlay construction

Our first simulations aim at validating our overlay construction algorithm.

Fig. 5 plots the popularity distribution of nodes, measured as the average number of different servants per minute (including the node itself) that a node can offer to querying users. Two overlay sizes $N = 5000$ and $N = 10000$ are considered to verify the scalability properties of the network3. The solid line represents a power law distribution (n) $\sim n^{-3}$, i.e., the node popularity distribution in a Barabási-Albert network. The figure shows how the EQUATOR overlay assumes a scale free topology which well approximates the Barabási-Albert network for both values of N. A certain discrepancy exists between EQUATOR and the theoretical curve for high popularity values. However, it is worth noticing how these differences are amplified by the log-log scale of the graph. Since values are related to very small portions of the entire overlay population, differences are actually of little significance. In EQUATOR, the overlay is dynamic [4] and hence links between nodes change frequently. Table I reports on the average clustering coefficient evaluated for different overlay sizes and compares it with the theoretical value of the Barabási-Albert network. We can observe how EQUATOR reasonably approximates the Barabási-Albert model also concerning this parameter, which is slightly higher than the theoretical value, but significantly lower than the clustering coefficient of highly clustered scale-free networks, e.g., the World Wide Web, whose clustering coefficient is about 0.1.

TABLE II

AVERAGE CLUSTERING COEFFIECIENT

Network Size	Equator	BA model
5000	0.0084	0.0036
10000	0.0072	0.0021

These results validate the overlay construction algorithm deployed in EQUATOR, as also confirmed by the results presented in the following.

C. Lookup performance

To validate the effectiveness of the EQUATOR overlay when providing lookup services, we consider the 1-hop average blocking probability (i.e., the probability that a user does not find an available servant when $D_l = 1$). Coherently with the assumptions of Section III-C, we consider a lookup hop to be exhausted when that node (that receives a service request) and all the servants it knows have been asked for the service.

We use as a reference the lookup performance obtained over a Barabási-Albert network where lookup procedures start only at nodes whose in-degree is greater than a given value M. We consider values for M ranging from 3 (corresponding to a percentage of nodes involved in the lookup procedures $p_{sp} = 16\%$) to 7 (corresponding to $p_{sp} = 5\%$) a good tradeoff between lookup performance and lookup load distribution among nodes, as discussed in Section III-E. Fig. 6 shows how EQUATOR[9][10] and this ideal network achieve comparable results.

In particular, EQUATOR behaves similar to a Barabási-Albert[10] overlay where $M = 5$ (corresponding to $p_{sp} = 8\%$). Given the limited size of caches in EQUATOR, this result is obtained thanks to the policies adopted in advertising peers and in handling such caches. These tend to favor the selection of popular nodes, thus approximating the behavior of a Barabási-Albert network where M assumes values reasonably greater than 1.

TABLE II

SAMPLE CUMULATIVEDISTRIBUTION FUNCTION OF THE LOOK UP/ADVERTISEMENT MESSAGE RECEIVED BY NODES

LOOKUP MESSAGES(PERCENTSGE OF MESSAGE)	PORTION OF NODES	ADV MESSAGES	PORTION OF NODES
≤0.01%	0.60	≤0.001	0.69
≤ 0.1%	0.62	≤0.01	0.76
≤ 1%	0.93	≤0.1	0.94
≤ 10%	0.99	≤1	0.98
≤ 100%	1	≤7	1

VI. CONCLUSION

This paper focuses on service-oriented overlays where users are interested to locate any of the many available overlay peers in the shortest time, i.e., the offered service is based on equivalent servants. Existing solutions, either structured or unstructured, can support these services but are not optimized for this purpose, which however is growing in importance due to the spread of many applications which need these specific features This paper compares structured and unstructured overlays, demonstrating through analytical and simulation results how an unstructured solution relying on a scale-free topology is an effective option to deploy for offering services based on equivalent servants. On the basis of this result, we proposed the Equivalent servAnt locaTOR (EQUATOR) architecture[7][8], which overcomes the issues related to the deployment of a scale-free topology for service location in a real network, mainly due to the static nature of the ideal scale-free construction algorithm and the lack of a global knowledge of the participating peers. Simulation results confirmed the effectiveness of EQUATOR, showing how it offers good lookup performance in conjunction with low message overhead and high resiliency to node churn and failures. Some possible future works are introduced in Section IV-F and are related to some complementary issues ranging from the proximity-aware selection of servants to the introduction of proper incentives to encourage nodes to join the EQUATOR overlay and offer their resources.

References

[1] V. Valancius, N. Laoutaris, L. Massoulie, C. Diot, and P. Rodriguez, "Greening the Internet with nano data centers," in . *Proc. ACM CoNEXT*, 2009

[2] "Tor: Anonymity online." Available: http://www.torproject.org

[3] "Skype: Free Internet telephony that just works." Available: http://www.skype.com

[4] P. Maymounkov and D. Maziéres, "Kademlia: a peer-to-peer information system based on the XOR metric," in *Peer-to-Peer Systems*, 2002,

[5] T. Klinberg and R. Manfredi, "Gnutella 0.6," June 2002. Available: http://groups.yahoo.com/group/the_gdf

[6]"Kazaa media desktop," 2001. Available: http://www.kazaa.com/

[7] G. P. Jesi, A. Montresor, and O. Babaoglu, "Proximity-aware superpeer overlay topologies," *IEEE Trans. Netw. Serv. . Manag.*, vol. 4, no. 2, pp. 74-83, Sep. 2007

[8] F. Hong, Y. Feng, M. Li, and Z. Guo, "Constructing incentive oriented overlay on mobile peer-to-peer networks," in *Proc. IEEE Int. Conf. Par. rocess.*,2007,p.52.

[9] J. Zhao and K. Xu, "Enhancing the robustness of scale-free networks," *J. Phys. A: Mathem. Theory*, vol. 42, no. 19, p. 195003, May

[10] R. Cohen, K. Erez, D. Ben-Avraham, and S. Havlin, "Breakdown of the Internet under intentional attack," *Phys. Rev. Lett.*,vol. 86, pp. 3682- 3685, Apr. 2001.

