International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 8, Issue 8 (September 2013), PP. 77-87

Equivalent Servants over P2P Networks

M. Vijaya Bhaskar[1], V.V. Sunil Kumar, M.Tech[2]
[1,2]PBR VITS, Affiliated to JNTUA, Ananthapuram

Abstract:- While peer-to-peer networks are mainly used to locate unique resources across the Internet, new interesting deployment scenarios are emerging. Particularly, some applica- tions (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 un- structured 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.

Index Terms:- Distributed services, equivalent servants, 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 and CPU cycles, the present emerging of the cloud computing paradigm 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.

At the same time, the current wave of distributed sharing services tends to involve resources available at the edge of the network and hence bases on the peer-to-peer (P2P) paradigm to achieve performance, scalability, and robustness. Among the possible examples, the *Desktop Grid* computing exploits unused resources (storage, computational power, etc.) available on widely located (home) computers, while *NaDa* [1] uses P2P technologies to build "Nano Data Centers" that exploit the DSL gateways placed in our homes. The idea is that users owning enough resources (e.g., a DSL gateway or a home-PC, which are unused for a great portion of time) may enter the cloud and start offering services.

Existing works lack in providing adequate support to these emerging distributed systems. In fact, most of them focus. In this context, a new set of services is emerging, where every servant is potentially able to satisfy users' requests. In fact, many operations delegated to the cloud (especially by thin clients) often require "limited" resources in terms of bandwidth, storage or CPU cycles, and therefore can be easily handled by any of the many peers participating in the abovementioned service-oriented overlays. We can say that these services are based on multiple, equivalent servants. As a few examples, we can cite the offloading of some computations that are too expensive for mobile devices, the localization of a relay required for anonymizing a communication (e.g., Tor [2]) or establishing a successful VoIP transfer (e.g., Skype [3]), the necessity to keep the state of users in an online game [4], or a Personal Video Recorder that temporarily stores TV streams when the user is offline, not to mention new online-based computational platforms (e.g., Google Chrome OS [5]). In this scenario, applications require the localization of an available servant (i.e., a node that is currently free and hence can offer the service) in the shortest time, rather than a precise resource.

On the development of a system supporting specific requests, ranging from a unique specific file to a set of resources characterized by well-defined parameters. While these systems can also support the localization of equivalent servants, they are not optimized for this purpose because of the different requirements they comply with, more stringent in terms of resource constraints, but simpler in terms of timely response. Hence, for example, they might be unable to locate a serving node in a very short time, such as a relay to be used in an incoming VoIP call. Furthermore, they may insert an unnecessary overhead in the servant lookup, due to the features they provide to support complex queries, which are of little help in the context of services based on equivalent servants.

This paper focuses on services provided by equivalent ser- vants and models and analyzes the performance of structured and unstructured 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. In particular, with the support of some analytical and simulation results, we show how an unstructured network based on epidemic dissemination and built over a scale-free overlay topology is an effective solution to deploy in this context. Then, we present EQUATOR (EQUivalent servAnt lo- caTOR), a P2P-based architecture deployable in real networks for the provision of services based on equivalent servants. EQUATOR aims at guaranteeing high lookup performance, as well as high robustness to failures and churn phases.

After a brief revision of the related work concerning the existing service-oriented overlays (Section II), the paper introduces some possible overlay architectures that can be adopted to support the location of equivalent servants and shows the benefits of scale-free networks in this particular context (Section III). Then, Section IV introduces EQUATOR and describes its operating principles. An extensive simulation study is presented in Section V to evaluate and validate the proposed solution. Finally, Section VI concludes the paper.

II. 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 (i.e., the capability of the system to provide a querying user with an available servant) offered by different architectures, presenting some analytical and simulation results which demonstrate that an unstructured overlay based on a scale-free topology is a good choice for handling our service. Then, we elaborate on the other interesting properties of this solution.

A. Structured overlays

We first investigate the possibility to deploy a structured overlay based on a general DHT, as it has been proposed in [19] for the P2PSIP architecture.

Since in our scenario all peers provide the same functional- ity (i.e., we have only one resource provided by many nodes), 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, [19] 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, 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 (which is a regular graph in existing DHTs) 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. For example, P2PSIP already uses a structured overlay to index all possible targets of a multimedia communication, i.e., all the user agents registered in the SIP domain. Using the same DHT to locate, if necessary, a relay node to support the communication (i.e., a servant among the many peers existing in the SIP domain) may be a considerable advantage for that application, which needs to maintain only one overlay structure that can be used for both functions.

B. Unstructured overlays

An efficient unstructured overlay 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 (e.g., how nodes spread information) 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* [21] encompassing a bounded number of nodes. Within this tech- nique, 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*.

The effectiveness of random walks depends on the overlay topology adopted in the system. Among other possibilities, a scale-free topology [22] may offer interesting features. In a scale free network, the node degree distribution follows a power-law $P(n) = cn^{-\gamma}$, where $P(n)$ 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 (e.g., a random graph or a regular topology) 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 (i.e., the hubs, which know a lot of possible servants) provides better performance than accurately distributing the requests among all nodes, as random solutions try to do. This extends the results obtained by Adamic et al. [23] in the context of traditional file lookups in P2P systems, which demonstrated the effectiveness of random walks in scale-free networks 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, 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. In essence, a large number of advertisement messages reach the hubs even with a small dissemination depth (namely, the number of hops encompassed by advertisement messages before elapsing) and a small out-degree (representing the number of peers to which a node directs advertisement messages).

Another interesting feature of scale-free networks is that they can scale to an arbitrarily large network size without modifying the degree distribution of nodes, which continues to follow the same law. This ensures that new hubs are automat- ically created when the network size grows, therefore main- taining the above described properties. In essence, scale-free networks potentially combine the advantages of centralized indexing (where a single entity directly handles all possible servants and consequently offers the best performance) and totally distributed solutions (which can scale to an arbitrary large number of participating servants and users).

One of the most popular mechanisms to build a scale-free network was proposed by Barabási and Albert [22] 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 characterized by a node degree distribution $P(n) = cn^{-3}$ and an average path length which behaves as $\ln N$ [22]. The Barabási-Albert model is used as a reference in the rest of the paper. Although in general $P(n) = P(m + d)$, in this case we are interested in the in- degree of a node as it represents its popularity, i.e., it counts the number of nodes that send their advertisements to it. Thereby, without losing in significance, we consider $P(n) = P(d)$ — i.e., the distribution of the in-degree of nodes — in the following.

The Barabási-Albert model is an ideal network formation algorithm that requires a global knowledge of the existing nodes. Clearly, this is not feasible in a real network. Hence, while this section shows the effectiveness of a scale-free solu- tion, Section IV will present an overlay construction algorithm based on a limited network knowledge which approximates the Barabási-Albert model.

III. EQUATOR

This section presents EQUATOR, an unstructured overlay based on the Barabási-Albert model (and hence on a scale-free topology), which adopts a set of construction and operating rules that are suitable for a real network. Furthermore, an epidemic dissemination algorithm is used to spread the net- work knowledge among the participating peers. A portion of a possible EQUATOR overlay is shown in Fig. 4

(some details will be clarified in the following), with some peers being part of the scale-free topology and some normal users accessing the offered service.

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 particular, the Barabási-Albert model requires a set of m_0 peers to be available at the initial step of the overlay setup. A simple solution (adopted in many existing overlays such as KaZaA [9]) consists in setting up some static peers and pre-configuring their addresses on each client.

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. Bootstrap servers globally store information about m0 participating peers; when a peer joins the overlay, it adds itself to the list in oase the number of entries in tbe bootstrap servers is n m0. Since entries in the bootstrap servers expire after a predeñned lifetime, each peer periodically ite-contacts the bootstrap servers and potentially adds itself to the list.

B. Nüde 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 deñne 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 irl-degree of the node. Since it is unfeasible for an EQUATOR node to be aware of its irl-degree, EQUATOR adopts as popularity metric the number of advertisement messages a node receives, which is proportional to its in-degree. In particular, a node can estimate its popularity by maintaining statistics about the average number of received messages per minute. The popularity of a node is used both in the overlay construction (to connect to the most popular nodes) and in the overlay maintenance (to keep connections to hubs) and is propagated in the advertisement messages

C. Overlay knowledge and advertisement

Each node in the overlay maintains two different node caches: a Servanr cache and an overlay cache. The former contains the set Si of Servants indexed by a peer 'Ui and it is populated by nodes that are lightly loaded with high probability, Le., 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 t0. Hence, it includes nodes of different popularity in Order to better represent the overlay. We denote by TSC the size of the servant cache and by TOC the size of the overlay cache.

At each advertisement round (which we suppose to occur every tad@ 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 t0 the preferential attachment mechanism. These messages contain a list of tuple <node, popularity, ngc entries are selected as the less popular peers present in the servant cache, while no@ 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 nsc entries in the servant cache and the HOC entries in the overlay cache. When caches are full, the ns.2 entries replace the most popular peers ofthe servant cache, while the entries replaced by the new noe 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 dislIibutions 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 Td > l, nodes along the dissemination path also insert themselves in the advertisement messages before forwarding the message to the next hop. Since these nodes are highly popular peers with high probability (for scale-free construction), this slightly biases the overlay cache with highlyI popular nodes, with the aim of favoring hubs to be contacted and hence promoting preferential attachment.

An example of the cache update process is shown in Fig. 4, when the EQUATOR peer receives an advertisement message from EP3 (the solid arrow in the figure). In the ůgure, a peer announces two peers il knows, one picked from the servant cache and one picked from the overlay cache (in addition to the node itseli). We also suppose a Cache size TSC = TOC = 4 peers and only one entry of each cache to be empty when the advertisement message arrives. The most popular peer of the servant cache and a randomly selected peer from the overlay Cache are removed to accommodate the newly discovered peers.

D. Cache refresh

In EQUATOR, the knowledge of the network at any time t is limited to a few nodes, Le., the ones that are in the two caches. Apparently, this is a radical departure from the Scalefree 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 trafñe (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.

Frequent entry refresh is also important for the overlay cache to allow the overlay to be dynamic and hence more robust. When a new peer joins, its overlay cache only contains the bootstrap nodes retrieved from the EQUATOR Bootstrap Service. Thanks to the refresh, nodes can insert new peers in their overlay cache and update the popularity information of the peers they already knows. This increasing knowledge of the network allows nodes to incrementally contribute to the construction of the scale-free topology as they can apply a more and more accurate preferential attachment. Hence, the overlay results in a scale-free topology, although variable over time. Furthermore, a frequent refresh ensures nodes are aware of live peers and hence well connected to the rest of the overlay.

E. Service lookup procedures from normal users

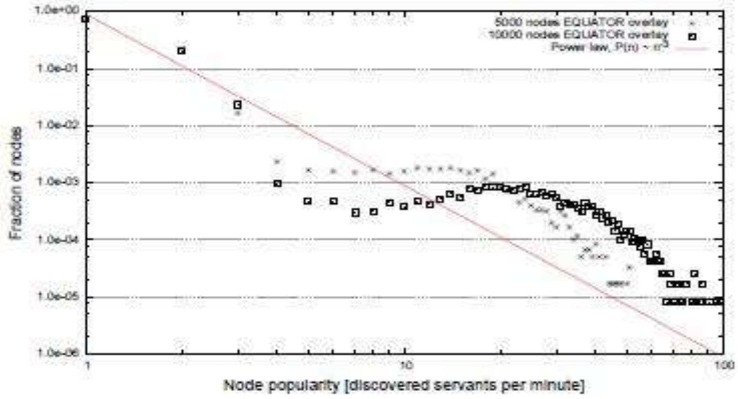
While the overlay contains all the peers that are available to offer some of their resources (i.e., are potential servants), many hosts may join the system as normal users in order to simply exploit the overlay services and without taking an active part in the overlay.

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. In fact, simply populating the lookup cache with nodes retrieved from the Bootstrap Service would possibly result in concentrating the lookup traffic among a few peers, with possible congestions.

IV. EQUATOR SIMULATION RESULTS

This section presents some simulation results on the EQUA- TOR architecture. We first validate our overlay construction algorithm, which we show to result in a scale-free topology. We also show how EQUATOR 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 pre- sented 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. Partici- pating 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. We model service requests with a further Poisson process. Whenever a service request is scheduled, it is randomly 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 Sec- tion III-C, the service duration is exponentially distributed. We consider several service request rates, ranging from 50 to150 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, either they are participating peers or users, contact this server and retrieve the m0 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 tadv = 30 min, which Section V-D will show to be proper values for the EQUATOR overlay. Moreover, we set nsc = noc = 3 nodes, m0 = 20 nodes. Finally, we set m = 2, Td = 2, and we assume that each peer can handle only one session at a time, as explained in Section III.

B. Overlay construction

Our first simulations aim at validating our overlay construc- tion algorithm. We assume node lifetimes to be exponentially distributed for simplicity, with an average node lifetime equal to 500 min. Different node dynamicity levels will be analyzed in the following.

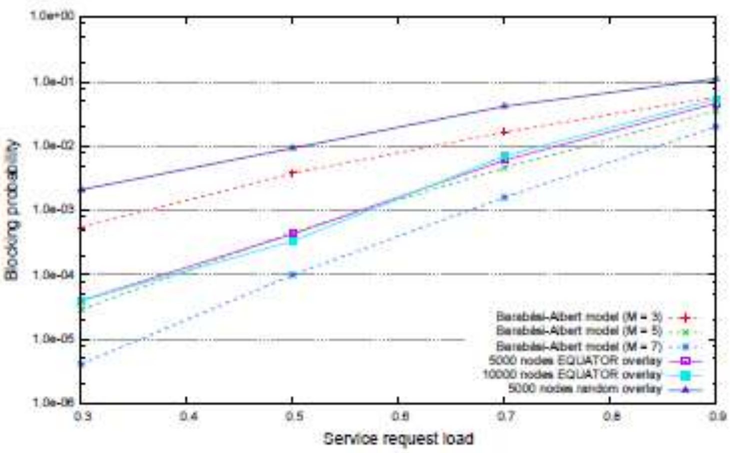


Fig. 5 plots the popularity distribution of nodes, measured as the average number of different servants per minute (including the node itself) that a node can offer to querying users. Two overlay sizes N = 5000 and

N = 10000 are considered to verify the scalability properties of the network[3]. The solid line represents a power law distribution $P(n) \sim n^{-3}$, 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 be- tween 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. Furthermore, they are mainly due to the difficulty in collecting adequate statistics because of the low number of nodes involved.

Besides the degree distribution, it is necessary to study the clustering coefficient of the EQUATOR network in order to complete the validation of our overlay construction algorithm. In EQUATOR, the overlay is dynamic and hence links between nodes change frequently. Consequently, we evaluate this pa- rameter as the average value among the clustering coefficients periodically observed in the network. We consider that, at a given instant of time, a node is connected to another if it sent an advertisement message to that node during the last advertisement round. Table I reports on the average clustering coefficient evaluated for different overlay sizes and compares it with the theoretical value [24] 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 [35].

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

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 trade- off between lookup performance and lookup load distribution among nodes, as discussed in Section III-E. Fig. 6 shows how EQUATOR and this ideal network achieve comparable results. In particular, EQUATOR behaves similar to a Barabási-Albert overlay where M = 5 (corresponding to p_{sp} = 8%).

Given the limited size of caches in EQUATOR, this result isvobtained 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. This is confirmed by the cumulative distribution of the average percentage of lookup messages per minute received by nodes when $D_l = 1$, presented in Table II for the case N = 5000. Although about 40% of participating peers are target of lookups from users, about 7% of nodes handle 99% of service requests, i.e., $p_{sp} \approx 7\%$, with a consequent high lookup performance. For the sake of completeness, Fig. 6 also considers the lookup performance of EQUATOR when nodes select their neighbors randomly among peers in the overlay cache and users start lookup procedures from a node selected randomly among peers they know. These mechanisms emulate the behav- ior of existing hierarchical overlays (e.g., KaZaA).

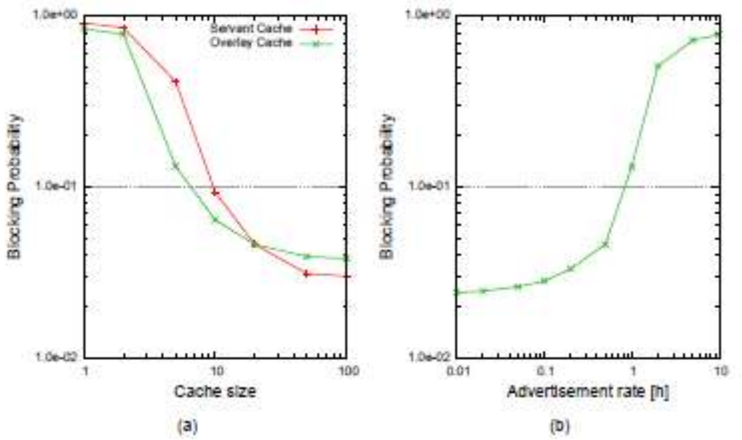
D. Effect of cache size and advertisement rate

In Fig. 7(a), we can observe how values of a few tens for τ_{sc} and τ_{oc} are sufficient to ensure a low blocking probability, which does not decrease significantly with a further increase of these values. In essence, a proper cache refresh, coupled with a limited cache size, allows EQUATOR to emulate an ideal system where each node has an arbitrary number of neighbors and a global knowledge of the network. To complete this analysis, Fig. 7(b) shows how an advertisement interval t_{adv} of a few tens of minutes is sufficient to ensure a good cache refresh. Lower values of t_{adv} are not necessary and do not provide a significant performance increase. This is due to the scale-free nature of the EQUATOR overlay: the shape and the short average path length it exhibits ensure a good refresh rate of the hub caches, thus leading to high lookup performance.

A higher advertisement rate may be necessary in order to use EQUATOR in different contexts, e.g., to locate specific resources. However, this is not the purpose of the system, which has been designed for locating equivalent servants. Adamic et al. [23] demonstrated the effectiveness of un- structured scale-free overlays when adopted to locate specific resources. However, this use of the scale-free topology requires different overlay maintenance, resource discovery, and lookup techniques that better support the offered service.

E. Message overheads

The above presented results prove the effectiveness of EQUATOR. In particular, they show how the scale-free topol- ogy ensures overlay efficiency with a limited advertisement rate ($t_{adv} = 30$ min), a small dissemination-depth ($T_d = 2$), and a limited cache size ($\tau_{sc} = \tau_{oc} = 20$ nodes). This results in a reduced per-node-overhead, as confirmed by Table II, which also includes the cumulative distribution of the average number of advertisement messages per minute processed at nodes when $N = 5000$. We can observe how 98% of nodes process less than 1 advertisement messages per minute and remaining 2% process always less than 7 messages per minute.



Concerning the lookup overhead, studied at the reasonable service request rate of 100 requests/min (i.e., $\rho_T = 0.6$) and for a network size $N = 5000$, we observed a maximum average service request rate at a single node of 5 messages/min. Fur- thermore, we observed a pick rate of about 20 messages/min, registered in about 1% of the total number of simulated minutes. This pick is mainly due to both the dynamics of request arrivals, which are modeled with a Poisson process. A hypothetical centralized solution would register an average request load on the central server of 100 messages/min (i.e., all requests would be directed to the server). This value is 20 times greater than the maximum average value observed in EQUATOR. Furthermore, also in this case we would register picks due to the characteristics of the request arrival process. When the network size grows, the network maintains its scale-free topology. Consequently, the number of nodes with an adequate popularity, which are likely to be contacted during lookup procedures, increases. Hence, although on equal load conditions the number of requests at the hubs increases, this value will not increase linearly with the size. For example, we also simulated a 50000 node overlay, where we did not see the maximum average request rate per node growing linearly from 5 to 50 messages/min. We registered instead a maximum value of 30 messages/min.

F. Failure probability

So far we considered the average blocking probability as a performance metric of EQUATOR, and compared it with the results obtained over a Barabási-Albert network. However, in a dynamic scenario such as EQUATOR, users can perceive service degradation also when an available servant is found, but then suddenly leaves the network before the service ends. This problem is common to all service-oriented overlays and can be mitigated in several ways, depending on the specific service deployed. Possible solutions are the utilization of backup nodes [20], the adoption of intelligent node selection and service migration policies [36], or the creation of appli- cation checkpoints [37]. The development of novel solutions in this context is outside the scope of the paper; however, we investigate for completeness how the EQUATOR architecture performs when different node lifetime distributions are used.

We consider three different network scenarios, characterized by different participating peer behaviors: a *highly dynamic overlay*, exemplified as a P2P-based Voice-over-IP network, a *moderately dynamic overlay*, exemplified as a P2P-based file-sharing network, and a *quasi-static overlay*, where partic- ipating peers are quasi-static nodes such as set-top-boxes, DSL gateways, data-centers, or various kinds of servers. Concerning the first scenario, the node lifetime distribution is obtained empirically after analyzing Skype traffic coming from/to the network of the University campus [38]. Node lifetimes are instead modeled as a Weibull distribution (shape = 0.2, scale = 1200) in the moderately dynamic overlay, as resulting in [39] for a file-sharing network. The third scenario is obtained by considering node lifetime exponentially distributed with an average node lifetime of 2 months (significantly longer than the average service duration). Table III reports on the overall failure probability (defined as the probability for the service to be disrupted, due to either a lookup failure or the servant node departure during the service exploitation) achieved in EQUATOR when $\rho_T = 0.6$. An overlay size $N = 5000$ is considered for these tests. Notice that the more dynamic the overlay, the higher is the failure probability, although backup nodes improve the overall performance. These results confirm how quasi-static nodes (such as the DSL gateways of NaDa or geographically distributed data-centers) are interesting poten- tial peers that can be used to build service-oriented overlays, and in particular EQUATOR.

In these tests we set $D_l = 4$, which allowed us to isolate the contribution of leaving servants from the overall failure probability, because the probability for a lookup to fail can be considered negligible (in fact, we did not observe lookup failures during simulations). Notice how this further confirms the effectiveness of overlay construction algorithm of EQUATOR as the system performs similarly to the Barabási- Albert network when $D_l > 1$ (see Fig. 3).

It is also interesting to investigate how node sudden and massive failures affect the overall failure probability. We defined a failure event in the EQUATOR simulator that pe- riodically replaces a given percentage p_f of peers (selected randomly) with new ones. Fig. 8 shows the evolution of the overall failure probability over time in the quasi-static scenario when $p_f = 0$ (i.e., no cancellation occurs).

As expected, the failure probability rapidly increases when the replacement occurs because the overlay topology is damaged and, conse- quently, lookups fail. However, also in this case the network automatically recovers in a reasonable amount of time. This is a major advantage of EQUATOR with respect to static scale- free networks and is due to both the policy adopted to populate the overlay cache and the dynamicity of links among nodes. The presence of lowly popular peers (which are not targets of the attack) in the overlay cache allows nodes to continue the advertisement and hence avoids the complete destruction of the network. This is in line with the theoretical results presented in [41], which demonstrates that the insertion of additional links among lowly connected nodes significantly increases the robustness of scale-free networks to hub deletions. The network dynamicity ensures nodes reconstruct the topology as highest popular peers are likely to be contacted during each advertisement round, thus further gaining in popularity and hence becoming the new hubs.

These results confirm the effectiveness of EQUATOR, which couples a high lookup performance with an adequate re- silience to failures and intentional attacks, even when massive node deletions occur.

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 (e.g., a proxy node to anonymize a communication). 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, 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 con- struction algorithm and the lack of a global knowledge of the participating peers. Simulation results confirmed the ef- fectiveness 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." [Online]. Available: http://www.torproject.org
[3]. "Skype: Free internet telephony that just works." [Online]. Available: http://www.skype.com
[4]. P. Bettner and M. Terrano, "1500 archers on a 28.8: Network program- ming in age of empires and beyond," in *Proc. Game Develop. Conf.*, 2001.
[5]. S. Pichai and L. Upson, "Introducing the google chrome os," 2009. [Online]. Available: http://googleblog.blogspot.com/2009/07/ introducing- google- chrome- os.html
[6]. I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17–32, 2003.
[7]. P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer informa- tion system based on the XOR metric," in *Peer-to-Peer Systems*, 2002, pp. 53–65.
[8]. T. Klinberg and R. Manfredi, "Gnutella 0.6," Jun. 2002. [Online]. Available: http://groups.yahoo.com/group/the_gdf
[9]. "Kazaa media desktop," 2001. [Online]. Available: http://www.kazaa. com/
[10]. *A Peer-to-Peer Approach to Resource Location in Grid Environments*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2002.
[11]. D. Puppin, S. Moncelli, R. Baraglia, N. Tonellotto, and F. Silvestri, "A grid information service based on peer-to-peer," in *Euro-Par*, 2005, pp. 454–464.
[12]. D. Tran, K. Hua, and T. Do, "A peer-to-peer architecture for media streaming," *IEEE J. Sel. Areas in Comm.*, vol. 22, no. 1, pp. 121–133, Jan. 2004.
[13]. T. Do, K. Hua, and M. Tantaoui, "P2vod: providing fault tolerant video- on-demand streaming in peer-to-peer environment," in *Proc. IEEE Int. Conf. Comm.*, vol. 3, June 2004, pp. 1467–1472.
[14]. *MAAN: A Multi-Attribute Addressable Network for Grid Information Services*, vol. 0, 2003.
[15]. A. R. Bharambe, M. Agrawal, and S. Seshan, "Mercury: supporting scalable multi-attribute range queries," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 353–366, 2004.
[16]. Y. Zhu and Y. Hu, "Ferry: A p2p-based architecture for content-based publish/subscribe services," *IEEE Trans. Par. Distrib. Systems*, vol. 18, no. 5, pp. 672–685, May 2007.
[17]. J. Albrecht, D. Oppenheimer, A. Vahdat, and D. A. Patterson, "Design and implementation trade-offs for wide-area resource discovery," *ACM Trans. Int. Technol.*, vol. 8, no. 4, pp. 1–44, 2008.
[18]. A. Awan, R. A. Ferreira, S. Jagannathan, and A. Grama, "Unstructured peer-to-peer networks for sharing processor cycles," *Parallel Comput.*, vol. 32, no. 2, pp. 115–135, 2006.
[19]. C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne, "Resource location and discovery (reload) base protocol," Internet Engi- neering Task Force, Internet Draft draft-ietf-p2psip-base-06, Nov. 2009, (Work in progress).
[20]. L. Ciminiera, G. Marchetto, F. Risso, and L. Torrero, "Distributed connectivity service for a sip infrastructure," *IEEE Network*, vol. 22, no. 5, pp. 33–40, 2008.
[21]. Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 30, no. 1, pp. 258–259, 2002.
[22]. R. Albert and A.-L. Barabási, "Statistical mechanics of complex net- works," *Rev. Mod. Phys.*, vol. 74, pp. 47–97, Jan. 2002.
[23]. L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in power-law networks," *Phys. Rev. E*, vol. 64, no. 4, pp.
[24]. 046 135+, Sep 2001.
[25]. K. Klemm and V. M. Eguíluz, "Growing scale-free networks with small world behavior," *Phys. Rev. E*, vol. 65, 2002, 057102.
[26]. J. Liang, R. Kumar, and K. W. Ross, "The kazaa overlay: A measurement study," *Comp. Netw.*, vol. 49, no. 6, 2005.
[27]. R. Cohen, K. Erez, D. B. Avraham, and S. Havlin, "Resilience of the internet to random breakdowns," *Phys. Rev. Lett.*, no. 21, pp. 4626–4628, Nov.
[28]. S. Voulgaris, D. Gavidia, and M. van Steen, "Cyclon: Inexpensive membership management for unstructured p2p overlays," *J. Netw. Syst. Manag.*, vol. 13, no. 2, 2005.
[29]. W. Yeager and J. Williams, "Secure peer-to-peer networking: The jxta example," *IEEE IT Profess.*, vol. 4, no. 2, pp. 53–57, 2002.
[30]. E. Tamani and P. Evripidou, "Applying trust mechanisms in an agent- based p2p network of service providers and requestors," in *Proc. IEEE Int. Symp. Cluster Comp. and the Grid*, 2006, p. 13.

[31]. F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: a decentral- ized network coordinate system," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 15–26, 2004.

[32]. 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.

[33]. 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. Process.*, 2007, p. 52.

[34]. G. Tan and S. Jarvis, "A payment-based incentive and service differen- tiation scheme for peer-to-peer streaming broadcast," *IEEE Trans. Par. Distr. Syst.*, vol. 19, no. 7, pp. 940–953, July 2008.

[35]. A. T. S. Ip, J. C. S. Lui, and J. Liu, "A revenue-rewarding scheme of providing incentive for cooperative proxy caching for media streaming systems," *ACM Trans. Multim. Comput. Commun. Appl.*, vol. 4, no. 1, pp. 1–32, 2008.

[36]. L. A. Adamic, "The small world web," in *Proc. Springer-Verlag Europ. Conf. Res. and Adv. Techn. for Dig. Libr.* London, UK: Springer-Verlag, 1999, pp. 443–452.

[37]. *Node selection for a fault-tolerant streaming service on a peer-to-peer network*, vol. 2. Los Alamitos, CA, USA: IEEE Computer Society, 2003.

[38]. R. de Camargo, F. Kon, and R. Cerqueira, "Strategies for checkpoint storage on opportunistic grids," *IEEE Distrib. Syst. Online*, vol. 7, no. 9, pp. 1–1, Sept. 2006.

[39]. D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi, "Detailed analysis of skype traffic," *IEEE Trans. Multim.*, vol. 11, no. 1, pp. 117–127, 2009.

[40]. V. Aggarwal, O. Akonjang, A. Feldmann, R. Tashev, and S. Mohr, "Reflecting P2P user behaviour models in a simulation environment," in *Proc. Euromicro Conf. Par., Distr. and Netw.-Based Process.*, 2008, pp. 516–523.

[41]. 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.

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