HAL
archives-ouvertes.fr

Synapse: A Scalable Protocol for Interconnecting Heterogeneous Overlay Networks

Luigi Liquori, Cédric Tedeschi, Laurent Vanni, Francesco Bongiovanni, Vincenzo Ciancaglini, Bojan Marinkovic

▶ **To cite this version:**

Luigi Liquori, Cédric Tedeschi, Laurent Vanni, Francesco Bongiovanni, Vincenzo Ciancaglini, et al.. Synapse: A Scalable Protocol for Interconnecting Heterogeneous Overlay Networks. Mark Crovella and Laura Marie Feeney and Dan Rubenstein and S. V. Raghavan. NETWORKING 2010 9th International IFIP TC 6 Networking Conference, Chennai, India, May 11-15, 2010. Proceedings, May 2010, Chennai, India. Springer Verlag, 6091, pp.67-82, 2010, Lecture Notes in Computer Science. <10.1007/978-3-642-12963-6_6>. <hal-00909544>

HAL Id: hal-00909544
https://hal.inria.fr/hal-00909544

Submitted on 26 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Synapse: A Scalable Protocol for Interconnecting Heterogeneous Overlay Networks*

Luigi Liquori[1**], Cédric Tedeschi[2], Laurent Vanni[1],
Francesco Bongiovanni[1], Vincenzo Ciancaglini[1], and Bojan Marinkovic[3]

[1] Institut National de Recherche en Informatique et Automatique, France
Email: firstName.lastName@sophia.inria.fr
[2] Université de Rennes I/INRIA, France
Email: Cedric.Tedeschi@inria.fr
[3] Mathematical Institute of the Serbian Academy of Sciences and Arts, Serbia
Email: bojanm@turing.mi.sanu.ac.rs

Abstract. This paper presents Synapse, a scalable protocol for information retrieval over the inter-connection of heterogeneous overlay networks. Applications on top of Synapse see those intra-overlay networks as a unique inter-overlay network. Scalability in Synapse is achieved via co-located nodes, i.e. nodes that are part of multiple overlay networks at the same time. Co-located nodes, playing the role of *neural synapses* and connected to several overlay networks, give a larger search area and provide alternative routing.
Synapse can either work with "open" overlays adapting their protocol to synapse interconnection requirements, or with "closed" overlays that will not accept any change to their protocol.
Results from simulation and experiments show that Synapse is scalable, with a communication and state overhead scaling similarly as the networks interconnected. thanks to alternate routing paths, Synapse also gives a practical solution to network partitions.
We precisely capture the behavior of traditional metrics of overlay networks within Synapse and present results from simulations as well as some actual experiments of a client prototype on the Grid'5000 platform. The prototype developed implements the Synapse protocol in the particular case of the interconnection of many Chord overlay networks.

Keywords. Peer to peer, overlay networks, information retrieval.

1 Introduction

Context. The inter-connection of overlay networks has been recently identified as a promising model to cope with today's Internet issues such as scalability, resource discovery, failure recovery or routing efficiency, in particular in the context of information retrieval. Some recent researches have focused on the design of mechanisms for building bridges between heterogeneous overlay networks for the purpose of improving cooperation between networks that have different routing mechanisms, logical topologies and maintenance policies. However, more comprehensive approaches of such interconnections for information retrieval and both quantitative and experimental studies of its key metrics, such as satisfaction rate or routing length, are still missing. During the last decade, different overlay networks were specifically designed to answer well-defined needs such as content distribution through unstructured overlay networks such

* Supported by AEOLUS FP6-IST-15964-FET Proactive and DEUKS JEP-41099 TEMPUS.
** Corresponding author. Thanks to Ernst Biersack for the precious discussions.

as Kazaa or through structured networks, mainly under the shape of Distributed Hash Tables [12, 13, 15], publish/subscribe systems [2, 10].

An overview of the current problem. Many disparate overlay networks may not only simultaneously co-exist in the Internet but also compete for the same resources on shared nodes and underlying network links. One of the problems of the overlay networking area is how heterogeneous overlay networks may *interact* and *co-operate* with each other. Overlay networks are heterogeneous and basically unable to cooperate each other in an effortless way, without merging, an operation which is very costly since it not scalable and not suitable in many cases for security reasons. However, in many situations, distinct overlay networks could take advantage of cooperating for many purposes: collective performance enhancement, larger shared information, better resistance to loss of connectivity (network partitions), improved routing performance in terms of delay, throughput and packets loss, by, for instance, cooperative forwarding of flows.

As a basic example, let us consider two distant databases. One node of the first database stores one $(key, value)$ pair which is searched by a node of the second one. Without network cooperation those two nodes will never communicate together. As another example, we have an overlay network where a number of nodes got isolated by an overlay network failure, leading to a partition: if some or all of those nodes can be reached via an alternative overlay network, than the partition "could" be recovered via an alternative routing.

In the context of large scale information retrieval, several overlays may want to offer an aggregation of their information/data to their potential common users without losing control of it. Imagine two companies wishing to share or aggregate information contained in their distributed databases, obviously while keeping their proprietary routing and their exclusive right to update it. Finally, in terms of fault-tolerance, cooperation can increase the availability of the system, if one overlay becomes unavailable the global network will only undergo partial failure as other distinct resources will be usable.

We consider the tradeoff of having one vs. many overlays as a conflict without a cause: having a single global overlay has many obvious advantages and is the *de facto* most natural solution, but it appears unrealistic in the actual setting. In some optimistic case, different overlays are suitable for collaboration by opening their proprietary protocols in order to build an open standard; in many other pessimistic cases, this opening is simply unrealistic for many different reasons (backward compatibility, security, commercial, practical, etc.). As such, studying protocols to interconnect collaborative (or competitive) overlay networks is an interesting research vein.

Contribution. The main contribution of this paper is to introduce, simulate and experiment with *Synapse*, a scalable protocol for information retrieval over the interconnection of heterogeneous overlay networks. The protocol is based on co-located nodes, also called *synapses*, serving as low-cost natural candidates for inter-overlay bridges. In the simplest case (where overlays to be interconnected are ready to adapt their protocols to the requirements of interconnection), every message received by a co-located node can be forwarded to other overlays the node belongs to. In other words, upon receipt of a search query, in addition to its forwarding to the next hop in the current overlay (according to their routing policy), the node can possibly start a new search, according to some given strategy, in some or all other overlay networks it belongs to. This obviously implies to providing a Time-To-Live value and detection of already

processed queries, to avoid infinite loop in the network, as in unstructured peer-to-peer systems.

In case of concurrent overlay networks, inter-overlay routing becomes harder, as intra-overlays are provided as some black boxes: a *control* overlay-network made of co-located nodes maps one hashed key from one overlay into the original key that, in turn, will be hashed and routed in other overlays in which the co-located node belongs to. This extra structure is unavoidable to route queries along closed overlays and to prevent routing loops.

Our experiments and simulations show that a small number of well-connected synapses is sufficient in order to achieve almost exhaustive searches in a "synapsed" network of structured overlay networks. We believe that Synapse can give an answer to circumventing network partitions; the key points being that: (i) several logical links for one node leads to as many alternative physical routes through these overlay, and (ii) a synapse can retrieve keys from overlays that it doesn't even know simply by forwarding their query to another synapse that, in turn, is better connected. Those features are achieved in Synapse at the cost of some additional data structures and in an orthogonal way to ordinary techniques of caching and replication. Moreover, being a synapse can allow for the retrieval of extra information from many other overlays even if we are not connected with. We summarize our contributions with the following: (i) the introduction of *Synapse*, a generic protocol, which is suitable for inter-connecting heterogeneous overlay networks without relying on merging in presence of open/collaborative or closed/competitive networks; (ii) extensive simulations in the case of the interconnection of structured overlay networks to capture the real behavior of such platforms in the context of information retrieval, identify their main advantages and drawbacks; (iii) the deployment of a lightweight prototype of Synapse, called `JSynapse` on the Grid'5000 platform[4] along with some real deployments showing the viability of such an approach while validating the software itself; (iv) finally, on the basis of the previous item, the description and the deployement on the Grid'5000 platform of a open source prototype, called `open-Synapse`, based on the `open-Chord`[4] implementation of Chord inter-connecting an arbitrary number of Chord networks. The final goal is to grasp the complete potential that co-located nodes have to offer, and to deepen the study of overlay networks' inter-connection using these types of nodes.

Outline. The remainder of the paper is organized as follows: In Section 2, we introduce our Synapse protocol, declined for open/collaborative overlays (*white box*) viz. closed/competitive (*black box*) overlays. We provide examples. In Section 3, we present the results of our simulations of the Synapse protocol to capture the behavior of key metrics traditionally used to measure the efficiency of information retrieval. In Section 4, we describe a deployment of a client prototype[5] over the Grid'5000 platform. Section 5, we summarize the mechanisms proposed in the literature related to the overlay networks' cooperation. Conclusions and further work conclude. Due to lack of space, the protocol pseudocode is presented in a separate web appendix[5].

2 The Synapse Protocol

Architecture and assumptions. We now present our generic *meta*-protocol for information distribution and retrieval over an interconnection of heterogeneous overlay

[4] http://www.grid5000.fr and http://open-chord.sourceforge.net.

[5] Code and web appendix are available at http://www-sop.inria.fr/teams/lognet/synapse.

networks. Information is a set of basic `(key,value)` pairs, as commonly encountered in protocols for information retrieval. The protocol specifies how to insert information (`PUT`), how to retrieve it through a key (`GET`), how to invite nodes in a given overlay (`INVITE`), and how to join a given overlay (`JOIN`) over a heterogeneous collection of overlay networks linked by co-located nodes. These co-located nodes represent a simple way to aggregate the resources of distinct overlays. We assume each overlay to have its own inner routing algorithm, called by the Synapse protocol to route requests inside each overlay. We assume no knowledge of the logical topology of all the involved overlay networks connected by Synapse. To ensure the usual properties of the underlying network, we assume that communication is both symmetric and transitive. Synapse simply ignores about how routing takes place inside the overlays, Synapse only offers a mechanism to route from one overlay to another in a simple, scalable and efficient way.

The inter-overlay network, induced by the Synapse protocol, can be considered as an aggregation of heterogeneous sub-overlay networks (referred to as *intra*-overlay networks henceforth). Each intra-overlay consists of one instance of, e.g., Chord or any structured, unstructured or hybrid overlay. We recall that an overlay network for information retrieval consists of a set of nodes on which the information on some resources is distributed. Each intra-overlay has its own key/value distribution and retrieval policy, logical topology, search complexity, routing and fault-tolerance mechanisms, so on and so forth. The Synapse protocol can be summarized by the following points:

- *Synapses:* the interconnection of intra-overlay networks is achieved by co-located nodes taking part in several of these intra-overlays, called synapses. Each peer will act according to the policy of each of its intra-overlays, but will have the extra-role of forwarding the requests to some intra-overlay it belongs to.
- *Peer's name:* every peer comes with a proper logical name in each intra-overlay; in particular, synapses have as many logical names as the number of networks they belongs to.
- *Keys mapping in peers:* each peer is responsible for a set of resources (`key,value`) it hosts. Since every intra-overlay has different policies for keys distribution, we could say that also the inter-overlay induced by Synapse also inherits homogeneous distribution among the intra- and inter-networks. As for peers, every key comes with a proper logical name peculiar to each intra-overlay.
- *Set of resources assigned to set of nodes:* all overlay protocols for information retrieval share the invariant of having a set of peers responsibles of a specific set of resources. This invariant allows for routing under structured, unstructured and hybrid networks: the rationale is simple: by construction, intra-routing is the one always responsible for its correctness, since Synapse just cares about overlay's inter-connection.
- *Network independency and message translation:* intra-network protocols are different by construction: as such, when a message leaves a particular network and enters another network, the first network loses control of the route of that message inside the second one.
- *Topology, exhaustiveness, complexity and scalability:* by construction, the inter-overlay network induced by the Synapse protocol belongs to the category of unstructured overlay networks, with a routing that is not exhaustive, even if Synapse can connect only overlays that guarantee exhaustivity. The same goes for the

routing complexity that can be upper-bounded only in the presence of precise and strong hypotheses about the type of intra-overlay networks. The same goes for scalability: a Synapse inter-network is scalable if all the intra-networks are scalable.

- *Loopy routing avoidance:* to avoid lookup cycles when doing inter-routing, each peer maintains a list of tags of already processed requests, in order to discard previously seen queries, and a TTL value, which is decreased at each hop. These two features prevent the system from generating loops and useless queries, thus reducing the global number of messages in the Synapse inter-network.
- *Replications and Robustness:* to increase robustness and availability, a key can be stored on more than one peer. We introduce a Maximum-Replication-Rate (MRR) value which is decreased each time a `PUT` message touches a synapse, thus replicating the resource in more than one intra-overlay. This action acts as a special TTL denoting how many overlays can traverse a `PUT` message.
- *Social primitives:* each peer implements autonomously a `good_deal?` policy. This is a social-based primitive aimed at making some important choices that may strongly influence the performance and robustness of the Synapse routing. In particular, such a primitive is intended to help the choice of whether or not to join another intra-overlay, invite or accept a peer to one of the overlays, or even create a new network from scratch. There is no best good deal strategy: for example, if one network wants to increase connectivity with other overlays, it can suggest to all peers to invite and join all interesting/interested peers: this can be especially useful in case of high churning of the intra-network in order to increase alternative routing-paths through the neighboring intra-networks.

"White box" vs. "black box" synapse protocol. As stated in the introduction, one important issue in interconnecting overlay networks is the ability of one overlay to potentially modify its protocol instead of only accepting that co-located nodes will route packets without any change in the protocol itself. This is a concrete backward compatibility issue, since many overlays already exist, and it is hard to change them at this point for many reasons (security, commercial, technological ...).

As such, we have developed two variants of the synapse protocol; the first *white box* variant, is suitable to interconnecting overlays whose standards are open and collaborative, meaning that the protocol and the software client can be modified accordingly. The second, *black box* variant, is suitable to interconnecting overlays that, for different reasons, are not collaborative at all, in the sense that they only route packets according to their proprietary and immutable protocol. The white box allows the adding of extra parameters to the current inter-overlay we are connecting, while the black box deals with those extra parameters by means of a *synapse control network*, i.e. a distributed overlay that stores all the synapse parameters that cannot be carried on by the overlay we are traversing.

White box synapse. The white box hereby presented is capable of connecting heterogeneous network topologies given the assumption that every node is aware of the additions made to existing overlay protocols. The new parameters used to handle the game over strategy and replication need to be embedded into the existing protocols, so does the unhashed key in order to be rehashed when a synapse is met. One important requirement of the Synapse white box protocol with respect to other protocols using hash functions is that the keys and nodes' addresses circulate *unhashed* from hop to hop. Hash functions have no inverse: once a sought key is hashed, it is impossible to retrieve its initial value, and thus impossible to forward to another overlay having a

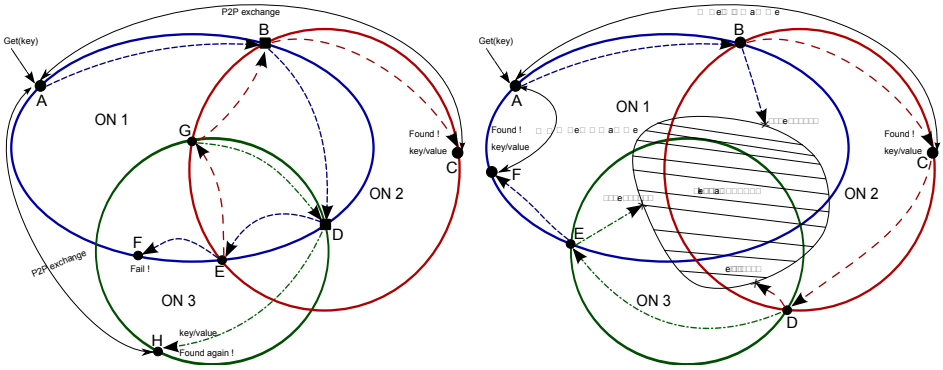

Fig. 1. Routing across differents overlays and dealing with a network partition

different hash function, since hash functions may vary (in implementations and keysize) from overlay to overlay. Both the hashed and the *clear* key data can be carried within the message, or a fast hash computation can be performed at each step. Standard cryptographic protocols can be used in case of strong confidentiality requirements, without affecting the scalability of the Synapse protocol itself.

Black box synapse. Interconnecting existing overlays made of "blind" peers, who are not aware of any additional parameters, seems to be a natural Synapse evolution and it constitutes a problem worth investigating. The assumption is that an overlay can be populated by blind peers (e.g. nodes previously in place) and synapses at the same time. Both interact in the same way in the overlay and exchange the same messages; moreover, those synapses can be members of several overlays independently (thus being able to replicate a request from one overlay to another) and can communicate with each other exclusively through a dedicated *Control Network* . The Control Network is basically a set of DHTs allowing each node to share routing information with other synapses without being aware of the routing of the undergoing message. So far the DHTs implemented are the following: (i) a Key table, responsible for storing unhashed keys circulating in the underlying overlays. Every synapse accessing this table can easily retrieve the key in clear way using only the information it is aware of; (ii) a Replication table, in which is stored the number of times the key should be replicated across all of the the overlays; (iii) a Cache table, used to implement the replication of `GET` requests, and cache multiple responses and control the flooding of foreign networks. Due to the obvious lack of space, the white and black box models are treated in the web appendix.

Example 1. Routing across differents intra-overlays. Figure 1 shows how a value present in one overlay can be retrieved from a `GET` launched by another overlay. Peer A in the overlay ON1 receives a `GET(key)` message: the routing goes until the synapse B, which triggers a second intra-overlay routing in ON2. The two routings proceed in parallel, and, in particular, the routing in ON2 terminates successfully with a peer-to-peer interaction between the peer A and peer C responsible of the resource. Routing continues on ON1 until synapse D, which triggers a third intra-overlay routing in ON3. The routing proceeds in parallel, and, in particular, routing in ON3 terminates successfully with a second peer-to-peer interaction between A and H, while routing in ON1 proceeds to a failure on peer F via the synapse E. Synapse E launches a fourth intra-overlay routing in ON2 that proceeds to a failure on node B (game over strategy) via synapse G. Finally, G launches a fifth intra-overlay routing on ON3, terminating

with a failure on D (again game over strategy). Peers playing game over strategy are depicted as squares.

Example 2. Dealing with network partition. Figure 1 also shows how intra-overlays take advantage of joining each other in order to recover situations where network partitioning occurs (because of the partial failure of nodes or the high churn of peers). Since network partitions affect routing performance and produce routing failures, the possibility of retrieving a value in a failed intra-overlay routing is higher, thanks to alternative inter-overlay paths. More precisely, the figure shows how a value stored in peer E of the overlay ON1 can be retrieved in presence of a generic network partition by routing via ON2 and ON3 through synapses B,C,D, and E. The reader can refer to the web appendix for a detailed description of the protocol pseudocode, in both the white and the black box model.

3 The Simulations

The purpose of the simulations is to allow for better understanding of the behavior of platforms interconnecting structured overlay networks through the Synapse approach. We focus on the key metrics traditionally considered in distributed information retrieval process, such as exhaustiveness (the extent of existing objects effectively retrieved by the protocol), latency (number of hops required to reach the requested object) and the amount of communications produced (number of messages generated for one request). We want to highlight the behavior of these metrics while varying the topology (the number of synapses and their connectivity, TTL, the number of intra-overlays ...).

Settings. Our simulations have been conducted using Python scripts, and using the *white box* protocol, capturing the essence of the Synapse approach. The topology of the overlay simulated is a set of Chord networks interconnected by some synapses. Information is a set of `(key,value)` pairs. Each pair is unique and exists once and only once in the network. We study the unstructured interconnection of structured networks. We used discrete-time simulation: queries are launched on the first discrete time step, initiating a set of messages in the network, and each message sent at the current step will be received by its destination (next routing hop) at the next hop.

Synapses. Our first set of simulations has the intent of studying how the previously mentioned metrics vary while we add synapses or increase the degree of existing ones (the number of intra-overlays a co-located node belongs to). The number of nodes was fixed to 10000, uniformly distributed amongst 20 overlays (approximately 500 nodes within each Chord). Queries are always triggered by one random node, the key sought by a query is also picked uniformly at random among the set of keys stored by the network. A query is said to be *satisfied* if the pair corresponding to the key has been successfully retrieved.

We first studied search latency, i.e. the number of hops to obtain the first successful response. As illustrated in Figure 2, one first point to notice is that the number of hops remains logarithmic when changing a Chord network into a Synapse network (the number of nodes is 10000, the latency never exceeds 14). Other experiments conducted by increasing the number of nodes confirm this. More precisely, Figure 2 (left) highlights the following behavior: (i) when the network contains only a few synapses, the latency first increases with the degree of synapses: only a few *close* keys are retrieved (keys available in the network of the node that initiated the query); (ii) then, when both parameters (the connectivity and the number of synapses) have reached a certain threshold, the searches can touch more synapses, and the whole

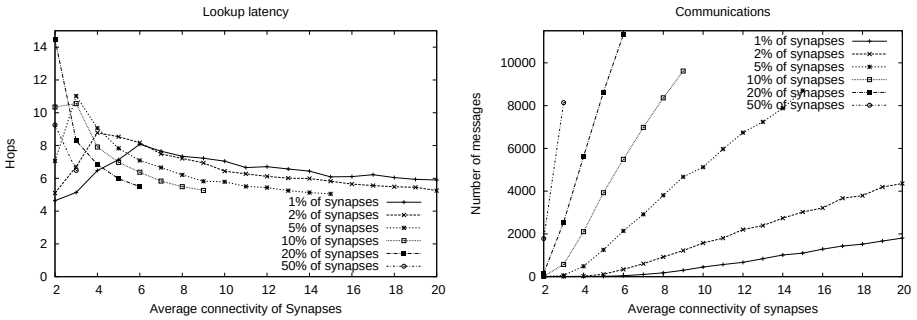

Fig. 2. Latency and communications in Synapse

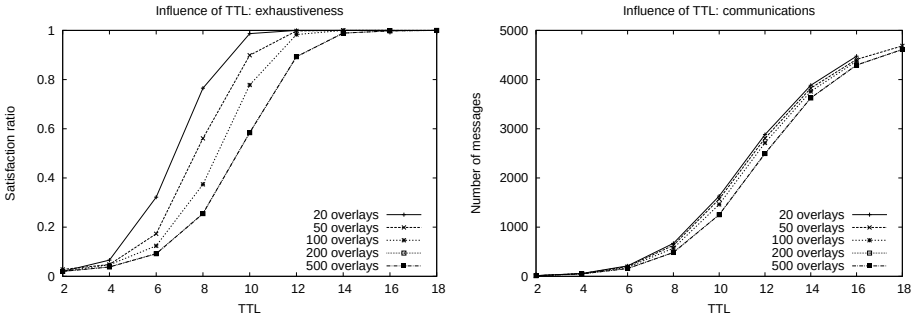

Fig. 3. TTL vs. exhaustiveness and communications

network becomes progressively visible, multiple parallel searches become more and more frequent and distant nodes (and keys) are reached faster. As we can see, increasing the number of synapses decreases the latency of only a small constant factor. In other words, synapse topologies does not need a lot of synapses to be efficient. This result fits with random graphs behavior: when the number of neighbors in the graph reaches a (small) threshold, the probability for the graph to be connected tends towards 1. Obviously, multiple searches in parallel lead to an increased number of messages. As illustrated in Figure 2 (right), this number increases proportionally with the connectivity and the number of synapses.

Time-To-Live. As we pointed out, the number of messages can become high when the number of synapses increases. To limit this impact, we introduced a Time-to-Live (TTL) to reduce the overhead while keeping an acceptable level of exhaustiveness. We launched a second set of experiments in order to study the impact of the TTL on the search queries. This TTL is simply decreased every time the query traverses a node.

The purpose is here is to preserve significant exhaustiveness, while reducing the amount of communications undergone by the inter-overlay. We made the number of overlays vary, to experiment the impact of the *granularity* of the network. In other words, a Synapse network made of few large structured intra-overlays could be called *strongly structured*, while another network with many smaller structured intra-overlays could be called *weakly structured*. The number of nodes was still set to 10000, and every node is a synapse belonging to 2 overlays chosen uniformly at random.

Figure 3 (left) confirms that a low synapse degree (2) is enough to achieve quasi-exhaustiveness. Another interesting result is that TTL can be bounded without any impact on the exhaustiveness (10 or 12 is enough even when the number of overlays

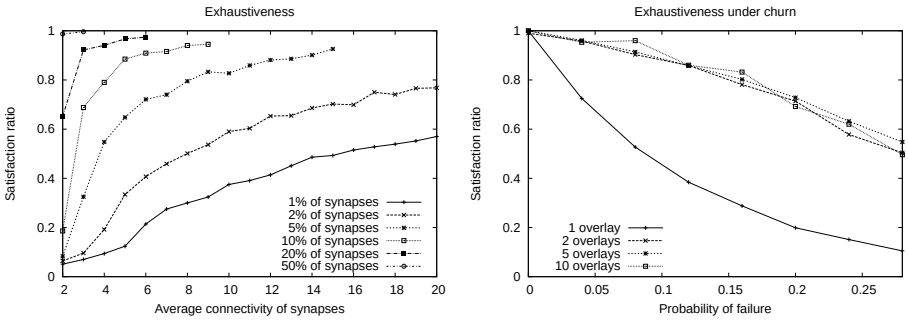

Fig. 4. Exhaustiveness vs. synapses and churn

interconnected is 500), while, as highlighted by Figure 3 (right), drastically reducing the amount of communications experienced, with the number of messages being almost divided by 2. To sum up, Synapse architectures can use TTL, leading to a significant exhaustiveness while drastically reducing the expected overhead. Finally, still see Figure 3, the *granularity* (defined above) does not significantly influence exhaustiveness and communications when the number and connectivity of the synapses are fixed.

Connectivity and Peers' churn Figure 4 (left) shows the evolution of the exhaustiveness while increasing the average number of overlays a synapse belongs to. We repeated the experiment for different ratios of synapses (in percentage of the total number of nodes). The exhaustiveness is improved by increasing both factors. We obtain more than 80% of satisfaction with only 5% of nodes belonging to 10 floors, and other nodes belonging to only one intra-overlay. When each node belongs to 2 overlays, the exhaustiveness is also almost guaranteed.

Since networks are intended to be deployed in a dynamic settings (nodes joining and leaving the network without giving notice), we conducted a final set of simulations to see the tolerance of Synapse compared to a single Chord overlay network. In other words, the question is *Does an interconnection of small Chords better tolerate transient failures than one large unique Chord?* In this experiment, at each step, a subset of nodes is declared unreachable (simulating the churn), making message routing fail. As we can see on Figure 4 (right), improvement on the number of satisfied requests can be obtained through a Synapse network: when the probability of failure/disconnection of a node increases, the global availability of the network is far less reduced with Synapse than with Chord. This shows that such synapse architectures are more robust and thus good candidates for information retrieval on dynamic platforms.

4 The Experimentations

JSynapse. In order to test our protocols on real platforms, we have initially developed JSynapse, a Java software prototype, which uses Java RMI standard for communications between nodes, and whose purpose is to capture the very essence of our Synapse protocol. It is a flexible and ready to be plugged library which can interconnect any type of overlay networks. In particular, JSynapse fully implements a Chord-based inter-overlay network. It was designed to be a lightweight easy to extend software. We also provided some practical classes which help in automating the generation of the inter-overlay network and the testing of specific scenarios. We have experimented with JSynapse on the Grid'5000 platform connecting more than 20 clusters on 9 different sites. Again, Chord was used as the intra-overlay protocol.

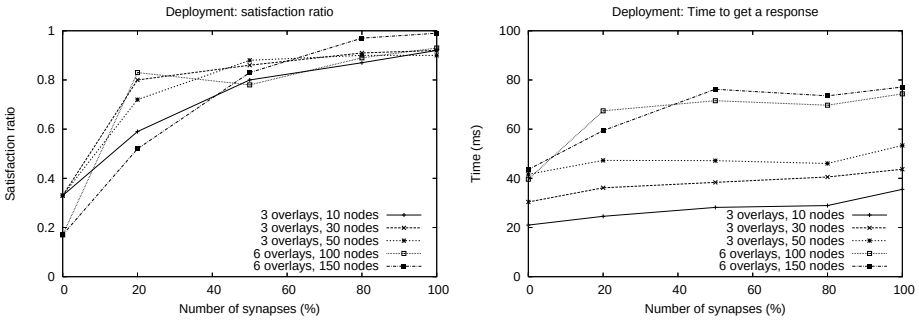

Fig. 5. Deploying Synapse : Exhaustiveness and Latency

We used one cluster located at Sophia Antipolis, France. The `Helios` cluster consists of 56 quad-core AMD Opteron 275 processors linked by a gigabit Ethernet connection. The created Synapse network was first made of up to 50 processors uniformly distributed among 3 Chord intra-overlays. Then, still on the same cluster, as nodes are quad-core, we deployed up to 3 logical nodes by processor, thus creating a 150 nodes overlay network, nodes being dispatched uniformly over 6 overlays. During the deployment, overlays were progressively bridged by synapses (the degree of which was always 2).

We give a proof of concept and show the viability of the Synapse approach while confirming results obtained by simulation. We also focus on the metrics affecting the user (satisfaction ratio and time to get a response). Once his request was sent, a user waits only for 1 second before closing the channels opened to receive responses. If no response was received after 1 second, the query is considered as not satisfied.

Figure 5 (left) shows the satisfaction ratio when increasing the number of synapses (for both white and black box versions). As expected, the general behavior is comparable to the simulation results, and a quasi-exhaustiveness is achieved, with only a connectivity of 2 for synapses. Figure 5 (right) illustrates the very low latency (a few milliseconds) experienced by the user when launching a request, even when a lot of synapses may generate a lot of messages. Obviously, this result has to be considered while keeping the performances of the underlying hardware and network used in mind. However, this suggests the viability of our protocols, the confirmation of simulation results, and the efficiency of the software developed.

Open-Synapse. We have also developed `open-synapse`, based on the stable and widely used `open-chord` implementation, which provides a complete and efficient Chord implementation. Open-Synapse extends `open-chord` core, thus taking advantage of its robustness and reliability. A preliminary set of tests on `open-synapse` involved 50 nodes and different randomly generated scenarii.

5 Related Work

Cooperation through hierarchy. While pointing out the limits of a unique global structured overlay network (rigidity, maintenance cost, security, etc.), several propositions have been made over the years to build alternate topologies based on the coexistence of smaller local overlay networks. The first approach has been based on hierarchical systems [6, 16], where some elected super-peers being are promoted to a top-level overlay network, leading to the requirement of costly merging mechanisms to ensure a high

level of exhaustiveness. In a more general view, merging several co-existing structured overlay networks has been shown to be a very costly operation [5, 14].

In the context of mobile ad hoc networks, Ariwheels [1, 10] has been designed to provide a large variety of services through a multi-layer overlay network, where super-peers, called Brokers, act as servers for a subset of peers. Ordinary peers, called Agents, submit queries to their Broker and receive results from it. Ariwheels provides an efficient mapping between physical devices in the wireless underlay network and virtual entities in the overlay network.

Cooperation through gateways.. Authors in [4] present two models for two overlays to be (de)composed, known as *absorption* (a sort of merging) and *gatewaying*. Their protocol enables a CAN-network to be completely absorbed into another one (in the case of the absorption), and also to provide a mechanism to create bridges between DHTs (in the case of the gatewaying). They do not specifically take advantage of a simple assumption that nodes can be part of multiple overlays at the same time, thus playing the role of natural bridges.

More recently, authors in [3] propose a novel information retrieval protocol, based on gateways, called *DHT-gatewaying*, which is scalable and efficient across homogeneous, heterogeneous and assorted co-existing structured overlay networks[6]. They argue that there is not one preferred structured overlay network implementation, and that peers are members of co-existing DHTs. Their assumptions are (i) only some peers support the implementations of different DHTs and (ii) some peers are directly connected to peers that are members of other DHTs, and are called *Virtual Gateways (VG))*. When a request is sent in one overlay, and no result was found, the requester can opt to widen his search by forwarding the original search request to nodes which belong to other structured overlay networks (mapping the search to the format supported by their relative overlay). A TTL value is added to the original search in order to avoid cycles; this value is decremented each time a request crosses a new DHT domain. Unfortunately the evaluation of their protocol lacks precious details and precision. It is unclear how they evaluate their protocol.

Cooperation through co-located nodes. Authors in [9] present Synergy, an overlay inter-networking architecture which improves routing performance in terms of delay, throughput and packet loss by providing cooperative forwarding of flows between networks. Authors suggest that co-located nodes are good candidates for enabling inter-overlay routing and that they reduce traffic. Our approach can also be seen as a deeper study of their concepts.

On the way of designing inter-overlay networking based on co-located nodes, authors in [8] present algorithms which enable a symbiosis between different overlays networks with a specific application in mind: file sharing. They propose mechanisms for hybrid P2P networks cooperation and investigate the influence of system conditions, such as the numbers of peers and the number of meta-information a peer has to keep. They provide interesting observations on how to join a candidate network, cooperative peers' selection, how to find other P2P networks, when to start cooperation, by taking into account the size of the network (for instance, a very large network will not really benefit from a cooperation with a small network), so on and so forth. Again, a more comprehensive understanding of this approach is missing.

[6] Ex. Two 160-bit Chord, or two 160/256-bit Chord, or one 160-bit Chord and one 256-CAN.

Authors in [7] consider multiple spaces with some degree of intersection between spaces, i.e. with co-located nodes. They focus on different potential strategies to find a path to another overlay from a given overlay, i.e. how requests can be efficiently routed from one given overlay to another one. They compare various inter-space routing policies by analyzing which trade-offs, in terms of state overhead, would give the best results, in terms of the number of messages generated and routed, the number of hops it takes to find a result and the state overhead (i.e. the number of fingers a node has to keep). They provide a comparative analytical study of the different policies. They showe that with some dynamic finger caching and with multiple gateways (in order to avoid bottlenecks and single points of failures) which are tactfully laid out, they obtain pretty good performances. Their protocol focuses on the interconnection of DHTs, while we extend it to any kind of overlays.

6 Conclusion

We have introduced Synapse, a scalable protocol for information retrieval in heterogeneous inter-connected overlay networks relying on co-located nodes. Synapse is a generic and flexible *meta*-protocol which provides simple mechanisms and algorithms for easy overlay network interconnection. We have given the set of algorithms behind our protocols and provided a set of simulations allowing to capture the behavior of such networks and show their relevance in the context of information retrieval, using key metrics of distributed information retrieval. We have also developed `JSynapse`, a lightweight implementation of Synapse, and experimented with it using the Grid'5000 platform, thus confirming the obtained simulation results and giving a proof of concept.

As future work, we first intend to focus on social mechanisms involved in synapses, which can greatly influence the shape of the network. On the practical side, we plan to extend `JSynapse` and plug in some other overlay network implementations. More intensive tests and deployments of `open-synapse`, our early prototype based on `open-chord` will also be considered.

References

1. D. Borsetti, C. Casetti, C. Chiasserini, and L. Liquori. *Heterogeneous Wireless Access Networks: Architectures and Protocols*, chapter Content Discovery in Heterogeneous Mobile Networks. Sringer-Verlag, 2009.
2. M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron. SCRIBE: A Large-Scale and Decentralized Application-Level Multicast Infrastructure. *IEEE Journal on Selected Areas in Communications (JSAC)*, 20, 2002.
3. L. Cheng. Bridging Distributed Hash Tables in Wireless Ad-Hoc Networks. In *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pages 5159–5163, 2007.
4. L. Cheng, R. Ocampo, K. Jean, A. Galis, C. Simon, R. Szabo, P. Kersch, and R. Giaffreda. Towards Distributed Hash Tables (De) Composition in Ambient Networks. *LNCS*, 4269:258, 2006.
5. A. Datta and K. Aberer. The challenges of merging two similar structured overlays: A tale of two networks. In *IWSOS*, 2006.
6. L. G. Erice, E. W. Biersack, K. W. Ross, P. A. Felber, and G. U. Keller. Hierarchical P2P Systems. In *Euro-Par*, 2003.
7. P. Furtado. Multiple Dynamic Overlay Communities and Inter-space Routing. *LNCS*, 4125:38, 2007.
8. K. Junjiro, W. Naoki, and M. Masayuki. Design and Evaluation of a Cooperative Mechanism for Pure P2P File-Sharing Networks. *IEICE Trans Commun (Inst Electron Inf Commun Eng)*, E89-B(9):2319–2326, 2006.
9. M. Kwon and S. Fahmy. Synergy: an Overlay Internetworking Architecture. In *Proc. of International Conference on Computer Communications and Networks*, pages 401–406, 2005.
10. L. Liquori, D. Borsetti, C. Casetti, and C. Chiasserini. An Overlay Architecture for Vehicular Networks. In *IFIP Networking, International Conference on Networking*, volume 4982 of *LNCS*, pages 60–71. Springer, 2008.
11. L. Liquori, C. Tedeschi, and F. Bongiovanni. BabelChord: a Social Tower of DHT-Based Overlay Networks. In *14th Symposium on Computers and Communications (ISCC 2009)*. IEEE, 2009. Short paper.
12. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Adressable Network. In *SIGCOMM*, 2001.
13. A. Rowstron and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-To-Peer Systems. In *Int. Conference on Distributed Systems Platforms (Middleware)*, 2001.
14. T. M. Shafaat, A. Ghodsi, and S. Haridi. Handling network partitions and mergers in structured overlay networks. In *Proc. of P2P*, pages 132–139. IEEE Computer Society, 2007.
15. I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup service for Internet Applications. In *ACM SIGCOMM*, pages 149–160, 2001.
16. Z. Xu, R. Min, and Y. Hu. HIERAS: A DHT Based Hierarchical P2P Routing Algorithm. In *ICPP*, 2003.