A Semantic Overlay Network for P2P Schema-Based Data Integration

Carmela Comito
University of Calabria
Rende, Italy
ccomito@deis.unical.it

Simon Patarin
University of Bologna
Bologna, Italy
patarin@cs.unibo.it

Domenico Talia
University of Calabria
Rende, Italy
talia@deis.unical.it

***Abstract*— Today data sources are pervasive and their number is growing tremendously. Current tools are not prepared to exploit this unprecedented amount of information and to cope with this highly heterogeneous, autonomous and dynamic environment. In this paper, we propose a novel semantic overlay network architecture, PARIS, aimed at addressing these issues. In PARIS, the combination of decentralized semantic data integration with gossip-based (unstructured) overlay topology management and (structured) distributed hash tables provides the required level of flexibility, adaptability and scalability, and still allows to perform rich queries on a number of autonomous data sources. We describe the logical model that supports the architecture and show how its original topology is constructed. We present the usage of the system in detail, in particular, the algorithms used to let new peers join the network and to execute queries on top of it and show simulation results that assess the scalability and robustness of the architecture.**

I. Introduction

Our always more connected world makes available to everyone an unprecedented volume of information. The surge of the Semantic Web, online bibliographic databases, file sharing networks, etc. are only few among the many examples of today's data sources. These data sources are characterized by their heterogeneity, and their dynamic and autonomic nature. In this context, semantic interoperability and source organization are key challenges to be addressed to allow information search, access and retrieval. In other words, one needs to be able to write queries and to execute them efficiently, over all available data sources.

In the past years, solutions have been proposed that address each of these problems independently. In the domain of semantic integration, formal integration models have been defined, query languages have been designed, schema mediation techniques have been proposed [1], [2], [3], [4], [5], [6]. These works however take little care of the underlying network whose topology is more than often abstracted. Similarly, peer-to-peer topologies [7], [8], [9], [10] have proven incredibly useful to manage and (self-)organize large networks of autonomic nodes. These topologies, however, do not incorporate any notion of semantics.

The PARIS (*Peer-to-peer ARchitecture for data Integration Systems*) semantic overlay network aims at filling this gap and proposes an integrated approach in which semantic data integration, based on schema mapping, and peer-to-peer topology are tightly bound to each other. The combination of decentralized data integration techniques [11] with gossip-based (unstructured) overlay topology management [8] and (structured) distributed hash tables (DHT) [9] enables rich queries to be performed on a number of autonomous data sources and makes their processing as efficient as possible. External efforts required to augment the system and to maintain it (even in the presence of failures) are kept to a minimal level, while still providing the level of flexibility, adaptability and scalability required to cope with the targeted highly dynamic environment.

The rest of this paper is organized as follows. Related work is discussed in Section II. The system model is presented in Section III. The design of a scalable and robust semantic overlay network that suits our design goals is depicted in Section IV. Then, Section V describes the functional architecture of PARIS. An experimental evaluation of the architecture, through simulation, is presented in Section VI and Section VII gives some concluding remarks, together with possible developments of this work.

II. Related Work

Peer-to-peer systems have been successfully used for sharing files described through simple attributes. Efforts have been done to refine topologies and query routing functionalities of these networks. Examples of such efforts are pure peer-to-peer systems, like Gnutella [12], based on flooding, or structures based on distributed hash tables as CAN [13] and Chord [9]. These systems only provide data sharing at file level, and a limited query language, usually based on file name search. Little effort has been made with respect to rich semantic representations of data and query functionalities beyond simple keyword searches. Only in the last few years, several peer-to-peer data management infrastructures allowing for semantic data sharing have been emerging [1], [2], [4]. All those systems focus on a decentralized integration approach (i.e. not based on a global schema): each peer represents an autonomous information system, and semantic data integration is achieved by establishing mappings directly among the various peers. Even if these systems achieve schema integration by adopting different formalisms, all of them abstract themselves from the underlying infrastructure and regard the system as a graph of interconnected data sources.

In PARIS, we exploit the complementarity of peer-to-peer overlay networks and peer-to-peer data management architectures to efficiently share large scale heterogeneous data, distributed over a large set of nodes. So, we combine peer-to-peer network topologies with appropriate query routing algorithms to ensure scalability and decentralized schema mapping to allow semantic interoperability. Some projects exist that rely on peer-to-peer overlay networks and offer rich semantic for data sharing. However, we differentiate from them by the design of an original topology which makes use of both structured and unstructured peer-to-peer overlay techniques. PIER [14] proposes an architecture for relational query processing with an index based on CAN. Differently from PARIS, PIER does not offer any data integration functionalities. Edutella [15] is a schema-based peer-to-peer network that applies the peer-to-peer architectural principle (using the primitives provided as part of the JXTA framework) to build a semantically enriched information system for the educational domain. The Edutella architecture is based on RDF to describe schemas and proposes efficient techniques for RDF query evaluation through a super-peer architecture. The global schema is replaced by a mapping network between local schemas that allows building new mappings transitively. Our approach is similar to Edutella in that it deals with different heterogeneous schemas in the network, and instead of using global schemas and correspondences, we have to rely on local transformation mechanisms and rules by using schema mappings. However, differently from Edutella our approach is completely decentralized in the sense that it does not rely on super-peers. In The Chatty Web [16], authors adopt a gossiping algorithm to dynamically map local schemas expressed by queries. In this model, the neighborhood of each node is composed of nodes containing the same schema or containing schemas with known mappings. A query is rewritten according to the mappings of the remote neighbor on which the query is propagated.

III. System Model

The primary design goal of PARIS is to develop a decentralized network of semantically related schemas that enables the formulation of queries over autonomous, heterogeneous and distributed data sources. The environment is modeled as a system composed of a number of peers, each bound to a data source. Peer schemas are connected to each other through declarative mappings rules. PARIS builds on the data integration model introduced for the XMAP integration framework [11]. The underlying integration model of this framework is based on schema mappings to translate queries between different schemas. Query translation is performed through the XMAP query reformulation algorithm. This algorithm receives as input an XPath query and produces as output zero, one or more reformulations of the query on the basis of the mapping rules related to the schema over which the input query is formulated.

Schemas, queries, and peers are given unique identifiers. Peer and query identifiers are "relative" to their schema. This means that these identifiers are, in practice, "prefixed" with their schema identifier.

A. Peers

We model our system as a collection $\mathcal{P}$ of *peers* which are logically bound to data sources. In other words, each data source D_p is represented by exactly one peer p and, conversely, each peer has access to a single data source, named *local data source*. Naturally, a *local schema* S_p is associated to this data source D_p. Each peer also holds a collection of mappings M_p from its local schema to other foreign schemas. Finally, a peer knows a list (also named *partial view* or, simply, *view*) of other peers (called *neighbors*). Beyond basic processing and communication facilities (exchanging messages with other peers), peers are supposed to be able to execute the above mentioned reformulation algorithm and to answer locally the queries they receive.

B. Groups

From the point of view of a single peer, the other peers in the network can be classified into four "groups". (1) The peer **local** group is made of all peers p_i that share the same schema. We note L_p the local group of peer p and $L_p = \{\ p_i \in \mathcal{P} \mid S_{p_i} = S_p\ \}$. Every peers in a local group L share the whole collection of mappings, denoted M_L, relative to the group schema S_L (the shared local group schema). (2) With respect to a peer local group L, the **semantic direct** group, D_L, is made of all local groups L_i with which a point-to-point mapping for the group local schema S_L is known. Formally: $D_L = \{\ L_i \in \mathcal{L} \mid S_L \rightsquigarrow S_{L_i}\ \}$, where $\mathcal{L}$ is the set of all known local groups, and $S_A \rightsquigarrow S_B$ denotes the existence of a direct mapping from schema A to schema B. The local groups of a semantic direct group share all their mappings, so the mapping collection, M_{D_L}, of D_L is composed by the union of the mappings of each local group L_i. (3) Again, with respect to a peer local group L, the **semantic transitive** group, T_L, is made of all local groups L_i whose schema is semantically related to the peer local schema S_L through a transitive mapping: $T_L = \{\ L_i \in \mathcal{L} \mid (\exists\ L_0, \ldots, L_k \in \mathcal{L}^{k+1} \mid S_L \rightsquigarrow S_{L_0} \wedge S_{L_0} \rightsquigarrow S_{L_1} \wedge \ldots \wedge S_{L_k} \rightsquigarrow S_{L_i})\ \}$. All members of a semantic transitive group share the same mapping collection M_{T_L} composed by the union of the mappings of each local group. (4) Finally, the peer **foreign** group is made of all remaining peers, i.e. all those that are not semantically related to the peer local schema. These notions will be illustrated in Figure 1 when we present the network topology.

IV. The PARIS Hybrid Topology

We have made the design choice to keep topology management and actual data integration at two distinct levels. This separation of concerns allows us to exploit recent algorithmic advances in the later domain on top of a scalable and robust overlay. We have chosen an hybrid topology for PARIS that mixes both kinds of overlays: structured and unstructured ones. More precisely, local groups are organized in unstructured overlays, while peers (or a large subset of them) are also part

of a DHT. This combination differentiates PARIS semantic overlay network from previously proposed architectures that have not taken advantage of both kinds of overlay at the same time [17], [18].

From the definition of local groups (see Section III-B), it is clear that all peers in the same local group share the same schema. What we want is the reverse to be also true: that all peers with the same schema are in the same group. It is therefore necessary to have strong guaranties that a group will remain connected even if a large number of peers fail. Gossip-based membership protocols are particularly well-suited for this task. We have chosen Newscast [10] to implement this protocol. Besides its conceptual simplicity, it offers excellent robustness and error-recovery properties.

In parallel, peers also participate in a DHT. From the way we construct peer identifiers (i.e. prefixed by the schema identifier), it is clear that all peers sharing the same schema will be contiguous in the identifier space. Our usage of the DHT will be limited to the capacity to send messages to a peer given its local schema. This means that we make use of the routing interface of the DHT, and ignore its storage capacities. We have chosen to use Chord [9] to implement this DHT. A property of Chord is that identifiers are integers and that data identifiers are always assigned to node whose identifier is immediately preceding in the live node set. With this characteristic in mind, the following steps must be performed to complete our task. First, we construct a random (data) identifier prefixed by the target schema identifier. We then send the message to the node responsible for this data using the standard DHT interface. The recipient node is either (i) a node with the same schema prefix, in which case we have succeeded, or (ii) the first node preceding, in the identifier space, the group we are interested in, in which case it is sufficient for this node to forward to its "successor" (this information is maintained by each node of the DHT as part of its normal behavior and this additional step is guaranteed to take exactly one hop). Note that the identifier used to make the lookup has to be chosen randomly so that it is not always the same peer in the group that is returned by this process.

We select only the peers that have the best performance (uptime, connectivity, etc.) within a local group to be part of the DHT. This selection will be the combined result of peer self-monitoring and explicit action taken by the administrator of the node (taken as a hint about the expected characteristics of the connection). Naturally, a minimum number of peers within a group is required for this selection mechanism to take place. The nodes who do not take part in the DHT will maintain a set of peer addresses (within their own local group) that will act as "gateways" to the DHT. This list is maintained in an epidemic manner as it is the case for the standard peer view.

The resulting semantic overlay network is depicted in Figure 1. In this figure, we assume that the following mappings exist: $S_A \rightsquigarrow S_C$, $S_C \rightsquigarrow S_B$, and $S_C \rightsquigarrow S_D$. From the point of view of the peer p we can classify the other peers as follows. The peer p belongs to the local group $L_p = A$ (i.e. whose shared local schema is S_A). Then, the local group A forms a semantic direct group with the local group C. The semantic transitive group of the local group A is composed of the local groups C, B and D. Finally, the local groups E and F are foreign groups for the peer p.

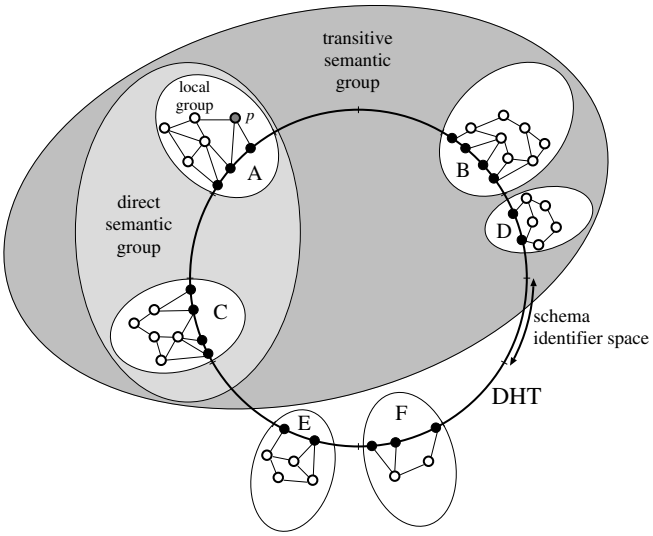

Fig. 1. PARIS semantic overlay network. Black nodes are peers with the best connectivity properties that are selected to participate to the DHT. White nodes willing to send messages to another group must hop through a black node in their local group. Groups are named from the point of view of A, assuming the following mappings exist: $A \rightsquigarrow C$, $C \rightsquigarrow B$, $C \rightsquigarrow D$.

V. Functional Architecture

This section discusses the functional architecture of PARIS. We have discussed topology issues in the previous section and will not explicit here the exact mechanisms required to maintain it. Details may be found in the original descriptions of these algorithms [8], [9].

Simply remind that, as part of the gossiping protocol, nodes regularly exchange messages with their neighbors, the neighbors belonging to the same local group as the peer itself.

A. Network Management

Letting nodes join (and leave) the semantic overlay network is obviously an important task to fulfil. We view this process as an iterative one, where the initial state is a network made of a single peer, to which nodes are added sequentially, one after the other.

For a node, joining the network means being inserted in its local group (the group corresponding to its local schema) and (possibly) the DHT. In order to do so, it is sufficient to provide the node with the address of any single peer in the system. This peer will take the incoming node schema identifier to locate a peer belonging to the corresponding local group using the DHT (possibly through a gateway), according to the technique described in Section IV. Here, we must distinguish two cases, whether this localization phase succeeds or not. (1) In the case of a successful localization, the incoming node obtains the

address of a peer that belongs to its local group. It will then use this "local contact" node to run the join protocol of the gossip-based overlay. The local contact (which is, by construction, part of the DHT) will decide next whether the incoming peer should be instructed to join the DHT or not. (2) A failed lookup means that there exists no other peers that share the same schema as the incoming node. As a consequence, the incoming node will form a new group by itself and its first contact will make it join the DHT (helped by a gateway, if necessary).

B. Query Processing

Processing queries submitted by the clients of the system is the main task of PARIS. A query may be submitted to any peer able to understand it, which means that queries submitted to any given peer must be expressed over its local schema.This condition holds for queries internally forwarded by the system. Upon reception of a query, each peer executes the following algorithm:

(1) Lookup the query identifier in the processed query table and drop it if it has already been processed. (2) Insert the query identifier in the processed query table. (3) Check whether the request has been submitted directly to this peer. If not, skip to the next step, else reformulate the query as follows. First, apply the reformulation algorithm to recursively produce reformulated queries expressed over all schemas semantically connected to the schema over which the query is formulated. In detail, this means: first find all local groups in the semantic transitive group of the local group to which the peer that has received the query belongs to. Then, on the basis of the available mapping rules, determine for which schemas among them it is possible to reformulate the given query. Finally, for each of the latter ones, produce one or more reformulations of the original query. Then, determine the associated local groups in the semantic transitive group of each schema over which one or more reformulations of the original query have been produced. Then, for each local group, use the DHT (possibly through a gateway) to send the reformulated queries to exactly one peer within the group according to the group local schema. (4) Broadcast the query to neighbors in the local group. (5) Execute the query locally. (6) Return the results to the originating client.

This algorithm is illustrated in Figure 2. We can see that the query Q_A is submitted to peer p (i). Q_A is reformulated by p into Q_C and, then, Q_B and Q_D (ii). The reformulated queries are sent to G, P's gateway, through the DHT (iii) and, from there, to peers in local groups B, C and D, respectively (iv). The original query is then broadcasted within the local group (v).

In this algorithm, contrary to the usual flood-based approach, the overlay is constructed in such a way that we know *a priori* that the neighbors of a given node are "interested" in the queries we forward to them only spurious messages exchanged are those sent to peers that have already seen the query. Even in this case, the processing overhead to reject such queries is minimal (a lookup in a table).

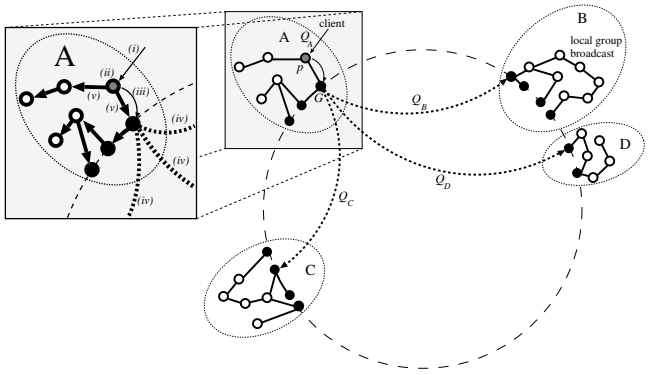

Fig. 2. Query processing in PARIS. In this example, we suppose that a query is submitted to a node of A and that we have the following mappings: $S_A \rightsquigarrow S_C$, $S_C \rightsquigarrow S_B$, $S_C \rightsquigarrow S_D$.

C. Mapping Management

In the steady state, we have said (Section III) that all peers within a semantic transitive group share the same collection of mappings. We must maintain this invariant when the network evolves. As for request processing, we assume that mappings are submitted to a peer whose local schema is the source of the mapping.

Mappings are manually crafted by administrators or users of the system. When a mapping is added to a running peer (or an existing one is modified), it is broadcasted to all the members of its local group. This mapping is also sent to one member of each local group in the semantic transitive group by using the DHT. These members will then broadcast this information to the other members of their respective local groups. This protocol is called the "new mapping" protocol.

When a node joins the network, if its local group is not empty, its local contact will provide it with the current knowledge of the group. Respectively, the incoming peer will run the above new mapping protocol for each mapping it knows which is not yet known by the group.

Finally, when the first node of a local group is inserted in the system, it will use the mappings it knows to contact other peers in its semantic direct group. The first node it finds will be used to run the new mapping protocol. It might be the case that no neighbors are found, for example when there exists some group for which no peer has joined the network. Mappings for which no peer has been found yet are flagged as such and reformulated queries are also sent to those "still empty" groups. When a peer is eventually found in a previously empty group, the "new mapping" algorithm is run with this peer and the reformulation algorithm is re-run to take the newly acquired information into consideration.

VI. Experimentation

We have designed PARIS with scalability and robustness in mind. In order to evaluate how our architecture supports these properties, we have developed a simulation of both the semantic overlay network architecture and the protocols of PARIS.

A. Simulation Environment and Setup

Our simulation of PARIS is based on the PeerSim framework [19], a Java based simulator specifically tailored for peer-to-peer protocol simulations. For the purposes of our simulation, we have implemented the routing protocol of Chord [9] and used an existing implementation of Newscast [10], a gossip-based membership protocol. Our implementation of Chord relies on the one proposed by the authors and includes all the refinements introduced after the original publication of the algorithm. We have also implemented a specific identifier assignment protocol to comply with the requirements of Paris identifiers (node identifiers are prefixed with a schema identifier). The default parameters that we have chosen are the following: network size $= 10^5$, number of schemas $= 500$, average mapping rank $= 2.0$, performance threshold for the nodes in the DHT $= 0.0$, gossip view size (or cache) $= 20$, Chord successors $= 5$.

B. Simulation Results

In this section we present the results we have obtained using our simulation environment. All numbers presented are averaged over 100 random queries.

1) Scalability: In a first series of experiments, we study the scalability of the PARIS architecture by observing the number of hops required to spread a query for execution over all nodes interested in that query.

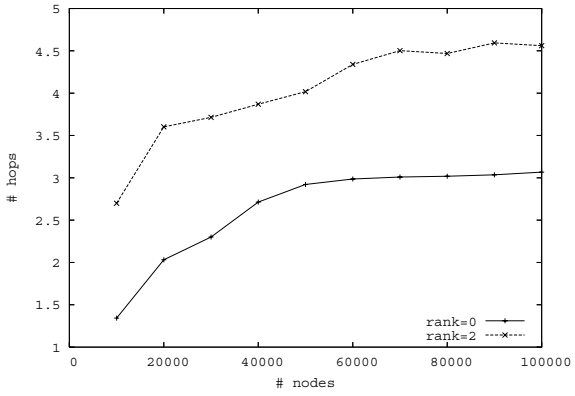

Fig. 3. Query span: maximum number of hops required to spread the execution of a query.

The Figure 3 shows the query span (the maximum number of nodes traversed in the longest branch of the processing tree, required to complete the execution of the query) as a function of the number of nodes in the network. The irregularities seen in the figure are due to the fact that the average number of nodes concerned by a specific query vary according to the network topology (when the rank is not null). In this figure, we can see that the span grows logarithmically with the size of the network and that it requires less than 5 hops at the maximum to spread a query over a 100 000 peer network. These results are particularly encouraging for the scalability of the architecture.

2) Chord: A concern with the PARIS architecture is its peculiar use of peer identifiers, in particular with respect to the DHT. In effect, most DHT assume that identifiers are evenly distributed in the identifier space, whereas PARIS identifiers are clustered by schemas and the number of schemas is orders of magnitude lower than the number of peers. However, lookups in PARIS are not randomly chosen: keys are derived from the same set of schema identifiers that are used for node identifiers. We performed another set of measurements to evaluate the influence of these parameters on the performance of the DHT.

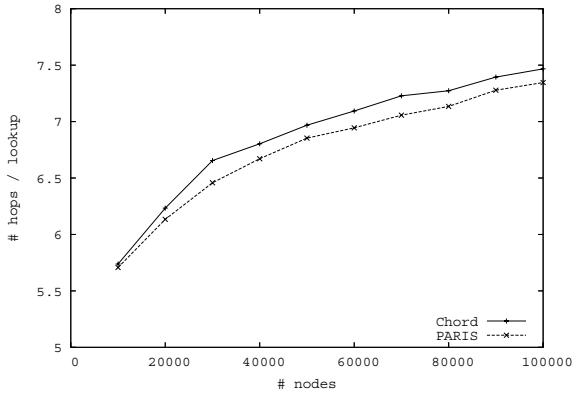

Fig. 4. Average number of hops by lookup w.r.t the size of the DHT.

The Figure 4 compares the average number of hops required to perform a lookup with an uniform identifier and key distribution (denoted "Chord" in the figure) and the number of hops used for PARIS distribution and lookups. What appears from this simulation is that, although being comparable, PARIS performance is in reality slightly better than the standard Chord.

3) Robustness: In order to assess the robustness of the architecture, we have performed a series of experiments where a large set of peers is removed, all at once. The initial network size is set to 100 000 peers and, at the end of cycle #3, from 10% to 90% of the network is brought down. In these experiments, the number of Chord successors has been increased to 10.

Figure 5 shows the average coverage for different DHT performance thresholds with an increasing proportion of nodes crashed. With a cache size of 40 peers, no connectivity loss is experienced for up to 50% of the nodes crashed. In itself this result is already quite encouraging: experiencing a crash of a half of the network is not something frequent. Still, using thresholds of 0.8 and more allows PARIS to support crashes of up to 80% of the whole network. This kind of massive failure is more likely related to network partitions and the results we have obtained show that if $1/5$ of the most available nodes become isolated from the rest of the network, they will continue to answer queries with all available resources.

The query span follows a relatively straight-forward evolution (Figure 6). The span increases right after the crash, as a result of peer caches being filled with dead nodes. However,

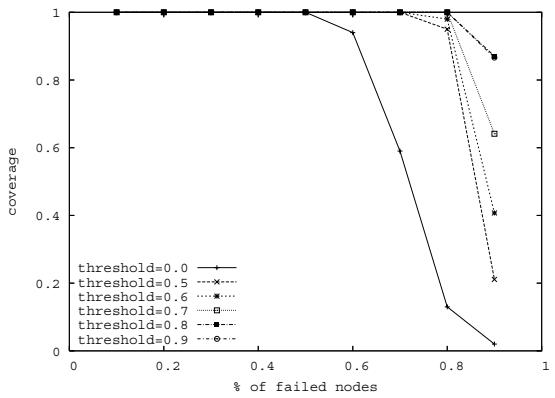

Fig. 5. Query coverage (proportion of nodes that have received a query among those that should have received it) in presence of catastrophic failures. Cache size is set to 40.

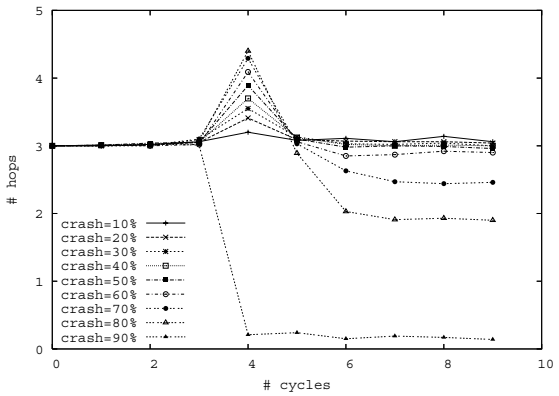

Fig. 6. Evolution of query span in presence of catastrophic failures. Cache size is set 20 and threshold to 0.5 (to rule out most connectivity issues).

in just a few cycles (at most 3), span stabilizes again and converges towards its expected value (according to the final network size). This obviously also depends on whether the connectivity was preserved or not. In the example shown in the figure, the connectivity is broken when 90% of the nodes are crashed. In this case, the final span value tends towards 0 as very few nodes may be reached during the normal query processing.

VII. Conclusion and Future Work

We have presented PARIS a semantic overlay network that enables data integration in a large-scale network of data sources. By using state-of-the-art technologies in both domains of data integration and peer-to-peer systems, we have built a system scalable, flexible and robust enough to cope with the exceptional characteristics of an environment such as the Internet. Building on an original hybrid topology, PARIS proposes an efficient query processing framework over a set of heterogeneous data sources. PARIS is still an on going work. We are currently investigating different algorithmic refinements to improve the local broadcast algorithm and balance the load over the peers more efficiently. We also plan to start the development of a software prototype and to deploy it on a testbed such as PlanetLab in the short term.

References

[1] Philip A. Bernstein, Fausto Giunchiglia, Anastasios Kementsietsidis, John Mylopoulos, Luciano Serafini, and Ilya Zaihrayeu, "Data management for peer-to-peer computing : A vision," in *WebDB 2002*, June 2002, pp. 89–94.

[2] Diego Calvanese, Elio Damaggio, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati, "Semantic data integration in P2P systems," in *DBISP2P 2003*, Sept. 2003, pp. 77–90.

[3] Enrico Franconi, Gabriel M. Kuper, Andrei Lopatenko, and Luciano Serafini, "A robust logical and computational characterisation of peer-to-peer database systems," in *DBISP2P 2003*, Sept. 2003, pp. 64–76.

[4] Alon Y. Halevy, Dan Suciu, Igor Tatarinov, and Zachary G. Ives, "Schema mediation in peer data management systems," in *ICDE 2003*, Mar. 2003, pp. 505–516.

[5] Anastasios Kementsietsidis, Marcelo Arenas, and Renée J. Miller, "Mapping data in peer-to-peer systems: Semantics and algorithmic issues," in *SIGMOD 2003*, June 2003, pp. 325–336.

[6] Sergey Melnik, Philip A. Bernstein, Alon Y. Halevy, and Erhard Rahm, "Supporting executable mappings in model management," in *SIGMOD 2005*, June 2005, pp. 167–178.

[7] Antony Rowstron and Peter Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Middleware 2001*, Nov. 2001, pp. 329–350.

[8] Ayalvadi J. Ganesh, Anne-Marie Kermarrec, and Laurent Massoulié, "Peer-to-peer membership management for gossip-based protocols," *IEEE Transactions on Computers*, vol. 52, no. 2, pp. 139–149, 2003.

[9] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, 2003.

[10] Màrk Jelasity, Rachid Guerraoui, Anne-Marie Kermarrec, and Maarten van Steen, "The peer sampling service: experimental evaluation of unstructured gossip-based implementations," in *Middleware 2004*, Oct. 2004, pp. 79–98.

[11] Carmela Comito and Domenico Talia, "XML data integration in OGSA grids," in *VLDB DMG'O5*, Sept. 2005, pp. 4–15.

[12] Mihajlo Jovanovic, Fred Annexstein, and Ken A. Berman, "Scalability issues in large peer-to-peer networks — A case study of Gnutella," Tech. Rep., University of Cincinnati, Jan. 2001.

[13] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Computer Communication Review*. Dept. of Elec. Eng. and Comp. Sci., University of California, Berkeley, 2001, vol. 31, pp. 161–172.

[14] Ryan Huebsch, Joseph M. Hellerstein, Nick Lanham, Boon Thau Loo, Scott Shenker, and Ion Stoica, "Querying the internet with PIER," in *VLDB 2003*, Sept. 2003, pp. 321–332.

[15] Wolfgang Nejdl, Boris Wolf, Changtao Qu, Stefan Decker, Michael Sintek, Ambjörn Naeve, Mikael Nilsson, Matthias Palmér, and Tore Risch, "EDUTELLA: a P2P networking infrastructure based on RDF," in *WWW2002*, May 2002, pp. 604–615.

[16] Karl Aberer, Philippe Cudré-Mauroux, and Manfred Hauswirth, "The chatty web: emergent semantics through gossiping," in *WWW2003*, May 2003, pp. 197–206.

[17] Peter A. Boncz and Caspar Treijtel, "Ambientdb: Relational query processing in a P2P network," in *DBISP2P 2003*, Sept. 2003, pp. 153–168.

[18] Alexander Löser, Felix Naumann, Wolf Siberski, Wolfgang Nejdl, and Uwe Thaden, "Semantic overlay clusters within super-peer networks," in *DBISP2P 2003*, Sept. 2003, pp. 33–47.

[19] M. Jelasity, Alberto Montresor, et al., "Peersim peer-to-peer simulator," Web site, 2005, http://peersim.sourceforge.net/.