DELIS
Dynamically Evolving, Large-scale Information Systems

Integrated Project
Member of the FET Proactive Initiative Complex Systems

Deliverable D2.3.2

Next-Generation Management Platform – Architecture, Algorithms and Prototypes
Start date of the project: January 2004
Duration: 48 months
Project Coordinator: Prof. Dr. math. Friedhelm Meyer auf der Heide
Heinz Nixdorf Institute, University of Paderborn, Germany
Due date of deliverable: December 2005
Actual submission date: January 2006
Dissemination level: PU – public

Work Package 2.3: Global management of competing transport services over large net domains
Participants: Telecom Italia (TILS), Italy
Universität Paderborn (UPB), Germany
Author of deliverable: Federico Morabito (federico.morabito@telecomitalia.it)
Fabrizio Davide (fabrizio.davide@telecomitalia.it)
Giovanni Cortese (g.cortese@computer.org)
Felix Heine (fh@upb.de)
1 Introduction

This document describes the work performed in Workpackage 2.3, which aims at a) investigating highly distributed architectural approaches, b) designing a collection of data management and coordination mechanisms, which can be applied to realize a management infrastructure for large collections of computing and communication resources (including transport networks, GRIDs, content distribution networks, etc). The work is based on the scenarios and requirements set in Deliverable 2.3.1. The management platform is a collection of tools and mechanisms addressing the requirements; a subset of such tools is actually implemented and integrated into a working software platform, while another subset consists of theoretical designs, which have been studied through simulation or analytical mechanisms. In brief, the main driver for research in WP 2.3 is to move from a centralized model of network management (where computing and communication nodes merely act as providers of information, and all decision making is performed at a central management station), to a fully distributed, peer-to-peer model; likewise, to move from a model requiring heavy involvement of human personnel in the management activities to one which realizes the ’self-*’ paradigm (self-configuring, self-repairing, self-optimising) for IT infrastructures. Methods for self-organization originated from P2P and complex network research are adopted and extended to reach the goal. Investigating novel ways for dissemination and search of network management data, is the first major objective of the WP. This has been the predominant area of research in second year of activity. By recognizing that network management is, to a large extent, a large-scale, decentralized data management problem, most of the work done so far has focused, according to the implementation plan, on researching data management techniques which apply to monitoring large collection of resources. A second research area for WP2.3, that we started targeting in second year and that will become prominent in 3rd year, is optimization of resource management. Preliminary work has started on techniques for understanding and controlling collaboration and competition among agents; this work is briefly summarized in Section 5. In this report, we first provide a brief summary of envisioned functionalities for the management platform in Section 2. Architectural approaches which were selected for investigation include a) structured overlay networks (DHT) and b) unstructured, random graphs with gossiping. Main architectural assumptions are summarized in Section 2. The majority of the work done is assuming structured overlays as underlying architecture. Research based on this architectural approach include:

- Research on mechanisms for content-based publish-subscribe and application level multicast complex queries.

- Research and implementation of storage and indexing mechanisms on DHT networks.

- Research and implementation of RDF-based storage and query mechanisms.

Unstructured networks are an interesting alternative architectural approach for information dissemination, although the challenges for achieving the needed reliability in practical settings are higher. Our research on unstructured networks and gossip-based approaches focuses on content-based publish subscribe.

Section 3 describes the individual research topics which have been investigated. Full technical reports are provided in the DELIS TR series and referenced from this deliverable.

Finally, as it came out from the activities during the first year, TILS and UPB have a common goal of researching innovative techniques for searching and monitoring information about heterogeneous resources distributed across the networks. We started the design of a joint prototype for handling network management data. The prototype platform is described in Deliverable 2.4.3.
2 Architectural Overview

This section aims at providing a high level summary of the next-generation management platform architecture. We describe the architectural framework which serves as a guideline to specific research activities. These activities using the framework are described in Section 3.

2.1 Architectural approaches

Architectural approaches which were selected for investigation include a) structured overlay networks (DHT) and b) unstructured, random graphs with gossiping.

2.1.1 Structured networks: DHTs

Structured P2P networks, specifically DHT-based P2P networks, are considered by many researchers as the overlay architecture which is more suited to support design of a variety of realistic P2P applications. The literature on DHTs is huge and we will not reiterate over it in this section. The specific research area which is relevant to WP 2.3 is related to P2P data management with DHTs. As a matter of fact, the application of structured overlay networks to support peer-to-peer database or heavily data-oriented applications, e.g. for implementing index structures which support complex queries, is an open research problem. Different approaches exist to address this goal. One architectural approach which we follow is based on creating a layered (on top of the DHT) index structure, which is exploited for query processing.

At the lower layer, the network of nodes (communication nodes e.g. routers, or computing/application nodes) whose management is the objective of the platform (the 'managed network'). Nodes communicate among them using e.g. IP protocols to achieve their primary functionality (e.g. routing in the case of transport networks); such communication network is also termed the 'underlay' in our work. Each node also participate to a management overlay network which is implemented as a P2P, DHT based network. The DHT realises a P2P network which provides communication and basic data retrieval services among all nodes in the managed network, and supports the implementation of management applications and generic management services. Data management facilities, including indexing and aggregation services are built on top of the DHT overlay (i.e. they use in general the DHT facilities for addressing and basic lookup of nodes and data items). All layers are designed as self-organising overlays; they support dynamic node arrival and departure. Also, they provide mechanisms for self-optimisation, which include distributed gathering of statistical information, and adaptation to current load of the system.

2.1.2 Unstructured networks

P2P unstructured overlay networks are considered as a viable solution for large scale data diffusion thanks to the fact that these networks try to keep their diameter as small as possible while maintaining the system connected despite continuous arrivals/departures of peers. Since their introduction, peer-to-peer (p2p) technologies were thought as possible solutions to introduce new forms of communication in wide, heterogeneous and dynamic communities, but the intrinsic dynamic behavior of the proposed model is a great obstacle to the building and maintenance of any logical structure on top of it, as nodes continuously entering and leaving the system disrupt this structure thus increasing the effort needed to maintain it. This disruptive effect can be so strong that, in extremely dynamic settings, even maintaining simple structures (like a connected graph) can become a non trivial problem. From this point of view unstructured p2p systems, that are based on random graphs, are considered more suited to such environments than structured ones. On the other side, such unstructured p2p systems offer only weak communication primitives to the developer, mainly based on broadcasting techniques. Such unstructured systems are built trying to approximate as much as possible a random
graph, as this can offer good qualities for broadcast-based communications, like low diameter and high connectivity. Some strategy must then be exploited to maintain such randomness in presence of churn and faults. From a theoretical point of view a random graph is built adding edges to a graph containing only vertices. Each edge is added with probability $p$. Building such a graph in a real system is a hard challenge. Nodes joining the system must build links to other nodes already in the system, and these links must be chosen randomly among all those that are possible. Realizing this without resorting to some centralized service can lead to non truly-random topologies, where not all links have the same probability of existence. To mitigate this problem, various works have proposed the usage of a technique known as view shuffling. With view shuffling every node periodically change its view (i.e. its set of link towards other nodes in the system) inheriting links from other nodes. The basic goal is to continuously change the system topology such that any possible link appear with the same probability, thus realizing an approximation of a random graph. This makes gossiping and flooding based techniques effective for applications that need to broadcast information (e.g., search/retrieve operations) networkwide.

Figure 1: Layered architecture.

However many applications need a more fine grained selection of the recipients of this information rather than an all/none option. Publish/subscribe communication paradigm provides this degree of selection of recipients based on the content of the information to be diffused.

### 2.1.3 Emergent approach to meta-data (schema) management

As an orthogonal concern to information dissemination, we research semantic interoperability among peers, leveraging approaches based on emergence. In current network management practice, agents exchange management information which must be structured according to standardized and a priori
agreed schema or ontologies. First, we are investigating ways for allowing the usage of semantically richer formalisms (with respect to the data models which are standard today such as SNMP and COPS) for expressing network management information. We study and devise efficient ways for storing and querying management information structured according to ontologies, described in RDF and possibly OWL, in a p2p data overlay. Also, we aim at devising mechanisms which allow a large and dynamic population of agents to exchange management information which need not be rigidly structured based on a single, centralized ontology (thus requiring 'ex-ante' agreements among all peers); instead, each peer is allowed (or more realistically, group of peers e.g. network boxes with same model, from the same vendor etc.) to understand a custom ontology. This requires that peers have the ability to perform mediation among the different schemas/ontologies when disseminating or querying information. This is a general issue in P2P data management, since one cannot rely on a single schema, describing data exported by all peers, which has been a priori designed. Semi-structured data models which allow more freedom in the specification of data, are useful. They support an emergent approach where the overall schema evolves over time. Schema extensions are done decentralized in a bottom-up fashion; the system takes care of the management of the related meta-data and provides the user query and browsing features which take this evolving meta-data into account. Again, data management for network monitoring requires schema evolution capabilities such as that outlined in previous paragraph. Current network management practice relies to a great extent to standardization of information exported by network elements. Standard schemas (termed Management Information Bases - MIBs - in network management lingo) define a small set of common information which any network element should export for interoperating with management applications. The limitations of the approach are well-known to network managers. Most routers (or other network objects, or applications) export a great deal of information using proprietary schemas, which can be accessed only by specialized applications. Supporting an evolutionary, 'emergent' approach to data management is therefore a major need for a next generation management platform. 

There are several research directions to be explored to achieve truly the goal of emergent semantics. We explored the relevant aspects, specifically we investigated effective strategies for query evaluation in p2p environments with semantically rich metadata based on RDF and RDF Schema. Our goal was to allow much more complex queries (arbitrary RDF graphs) which are evaluated w.r.t. RDF Schema semantics while maintaining and enhancing the scalability of the network. Because completeness is an important issue, we are focusing on RDF Schema / OWL-Lite rather than on DL based solutions.

3 Algorithms

In this section we present a summary of research activities on specific design approaches and algorithms. Full descriptions of the algorithms and designs are provided in the DELIS TR series.

3.1 Data dissemination via publish-subscribe communication

A generic pub/sub communication system (often referred to in the literature as Event Service or Notification Service) is composed of a set of nodes distributed over a communication network. Clients to the systems are divided according to their role into publishers, which act as producers of information, and subscribers, which act as consumers of information. Clients are not required to communicate directly among themselves but they are rather decoupled: the interaction takes place through the nodes of the pub/sub system. This decoupling is a desirable characteristic for a communication system because applications can be made more independent from the communication issues, avoiding to deal with aspects such as synchronization or direct addressing of subscribers from publishers. There are several architectural options and subscription models for publish-subscribe communication. Our reports [DELIS-TR-0228] and [DELIS-TR-0229] offer a complete survey. We focus on content-based
publish subscribe, and we explore two architectural options for realizing the broker overlay, namely peer-to-peer structured and peer-to-peer unstructured overlays.

### 3.1.1 Content-based publish-subscribe DHT networks

Content-based Publish/Subscribe (CBPS) interaction paradigm is suitable for a variety of large scale dynamic applications: news delivery, stock quoting, air and metropolitan traffic control, on-line games, dissemination of multimedia contents, dissemination of auction bids, services and resources discovery, and, more specifically to this initiative, remote monitoring of critical infrastructures and management of large scale systems. In contrast to their flexibility and expressiveness scalable CBPS systems are difficult to implement and the proposed solutions are not yet mature. In particular, current best accepted approaches are built over specialized overlays of message brokers. In contrast, there are no satisfying designs of CBPS over a DHT overlay. As outlined in the architectural section, we believe desirable leveraging DHT overlay in order to reuse its self-organization capabilities when developing higher level services. Plaxton routing provides a straightforward and convenient way to implement simple, topic-based publish subscribe (e.g. those exploited in systems such as Scribe, a publish-subscribe mechanism built over a Pastry DHT). However, more sophisticated subscription patterns cannot be implemented in a DHT network. The naf approach, which can be used to implement a pub-sub infrastructure over DHTs, requires an application developer to statically partition the set of events, and map partitions to different multicast groups. This approaches implies likely inefficiencies in bandwidth utilization, which are caused by unnecessary transmission of events (false positive, duplicate events). We propose a novel adaptive content-based subscription management system, relying on a Distributed Hash Table routing infrastructure. The approach is adaptive in that multicast groups are built and maintained by the pub/sub infrastructure so as to minimize unnecessary message transmission and bandwidth utilization; groups are periodically updated to rebalance traffic and keep up with the subscriptions of a changing population of peers. First, we define a model for the event space which is expressive enough for being used in any application domain. Then we provide mechanisms to dynamically identify groups of peers with similar subscriptions/preferences, and map them to a multicast group. Two different approaches for multicast group formation are proposed and discussed: a simple one, based on regular grid partitioning of the event space, and a more complex one based on clustering of users’ preferences. When the system evolves (e.g. joining and leaving of publishers/subscriber and/or node failures) multicast groups are reconfigured to maintain the desired level of system efficiency. System adaptation is supported by a scalable and dynamic application level multicast infrastructure, and by a run time performance measurement mechanism. The report [DELIS-TR-0228] gives a formal description of our solution and the main design guidelines. We also discuss advantages and disadvantages of our proposal, compared with the existing solutions.

### 3.1.2 Content-based publish-subscribe over random networks

The core mechanism behind a distributed pub/sub system is event routing. Informally, event routing is the process of delivering an event to all the subscribers that issued a matching subscription before the publication. We analysed and compared 3 different approaches to event routing:

- Flooding algorithms (event and subscription flooding).

- Selective algorithms (rendezvous-based and filter-based).

- Event gossiping algorithms (basic and informed gossiping).
We studied approaches based on unstructured overlay networks, here in the following described.

Unstructured P2P systems [PRU01, ADH05, EGH+03, GKM03] organize overlay nodes in a random graph characterized by some desired properties like low-diameter, high connectivity, constant-node degree. Each node in the overlay network maintains a partial view of the whole system containing network addresses of other nodes participating to the same network. The union of all the nodes together with their partial views, constitutes the whole overlay. Due to the large scale of such systems, partial views are usually limited in their size to a constant degree or to a size that varies only logarithmically with the overlay size.

We analysed existing approaches to overlay maintenance, with special emphasis on SCAMP and Cyclon. An evaluation through simulation is included in the report.

Cyclon [VGvS05] follows a proactive approach, where nodes perform a continuous periodical gossiping activity with their neighbors in the overlay and does not react to failures or departure of other nodes. Only joins are obviously managed in a reactive manner. The periodical gossiping phase (named “shuffle”) has the aim of randomly mixing the views between neighbor nodes. This provokes the long term effect of an overlay whose topology approximates a random graph.

The Cyclon approach has been selected as the basis for a novel pub/sub mechanism using an unstructured P2P as substrate.

The main problem in a publish/subscribe system is how to correctly identify the set of subscribers target for each specific event. From an abstract point of view, this problem is solved building for each event a list containing pointers (e.g. the IP address or other form of identification) to all the target processes. In the following we will refer to this list as the event distribution list (EDL).

In our work we study how to store Event Distribution Lists in such a setting. We compare a cooperative and a non cooperative model for storing EDLs, and suggest that the non-cooperative model (where only publishers and subscribers for an event collaborate to EDL storage and retrieval) is more suited to obtain needed accuracy in a dynamic system. We explore an approach based on subscription-flooding (SF), which disseminates subscriptions to all peers in the system. The approach, while not providing guarantee of fully accurate event distribution in a dynamic system (i.e. where nodes can freely join and leave), adopts several techniques to improve on the accuracy. Note that publishers are expected to disseminate advertisements which provide summary description of the class of events they generate to improve performance. Also, the algorithm leverages a view shuffle mechanism used to maintain the unstructured overlay, to avoid broadcasting subscriptions.

Subscription diffusion is realized through this simple protocol:

- each node sends its subscriptions and its advertisements to all the nodes in its view every time a view shuffle operation has completed;

- when a node receives a subscription from another node it puts it in a specific list called subscription history;

- when a node receives an advertisement from another node it matches this advertisement against its subscriptions and those contained in its subscription history. For each positive match it sends the address of the corresponding node to the node it received the advertisement from.

Aim of this protocol is to correctly keep distribution lists updated incurring less overhead than with a standard periodic broadcast. In fact our protocol requires communication only among neighbour nodes and can reach the same results obtained with periodic broadcast thanks to the fact that views change randomly and periodically.
3.2 Data dissemination and query processing algorithms for semi-structured data in p2p environments

In this work, we focus on the dissemination and querying of semi-structured RDF [MM04] and RDF Schema [BG04] data in p2p environments. In p2p data management systems, different system models are possible. Either each peer is treated as an individual unit, which can be queried and which will be deliver query results based only upon its own information. Above this layer, the network can employ query routing mechanisms, mediation, or distributed query processing. An alternative approach, which we follow in this section, is to query the aggregated network knowledge as a whole. In combination with appropriate semantics like RDF Schema or OWL [SWM04], this supports an emergent approach where each peer can contribute small pieces of information which are combined to build global, network-wide knowledge. However, from a technical point of view, this vision is difficult to achieve without loosing scalability. As we want to be able to combine arbitrary pieces of information, it is difficult to know a priori which peer might have some information which is useful for the current query. Thus we need techniques to distribute the data to well-known peers on a very fine-grained level to incorporate it in the query process. The resulting system is especially suited for the exchange and evaluation of meta-data. Thus it is not an alternative approach XML based p2p data management, but rather complements it. Our vision is a combined system which uses XML for data storage and RDF-based meta-data using the global data system model.

In our work, we provide a formal definition of the problem. We then focus on the distribution of RDF/S knowledge using a DHT based P2P network. This knowledge is the basis for query evaluation. As RDF is a graph-based knowledge representation, the basic operation of query evaluation is graph matching. However, as the knowledge is distributed over multiple nodes, we have to collect relevant parts of the graph prior to the final evaluation. We have developed and evaluated various strategies for the collection of the candidates and query evaluation.

The proposed design has been implemented using a Pastry [RD01] substrate, and is going to be integrated with the management platform developed at TILS to function as its metadata (schema) management component. The full description is provided in [DELIS-TR-0208] and [DELIS-TR-0316].

4 Main contributions

The main contribution to section 3.1.1 is described in [DELIS-TR-0228]. Also [DELIS-TR-0229] contributed to this work.

The main contribution to section 3.1.2 is described in the technical report Publish/subscribe for Dynamic Systems [DELIS-TR-0317]. This work is fully granted by DELIS project.

Finally, details for section 3.2 can be found in [DELIS-TR-0208]. Further advancements to the algorithms are presented in [DELIS-TR-0316].

5 Interim report on ongoing work on cooperation/ competition mechanism

As requested in the implementation plan, we report here on ongoing activities on cooperation/competition mechanisms. Most activities in this area so far have been performed in cooperation with SP5, and have as their scope the investigation of cooperation / competition mechanisms in the application scenario of content distribution. In this scenario, a set of content servers (CS) aim at serving their content to clients. Each content server can host replicas of content from other CSs in addition to their own content resources. A replication protocol operates in the background, through which CSs can request other CSs to store replicas of their content, to help them achieve their own service level (a good response time for their own content). CSs can adopt different strategies (from
fully selfish to fully altruistic) in deciding whether to serve or deny replication requests. We defined a protocol for such cooperative resource sharing, which we will implement in a simulation model to investigate the effects of different cooperation strategies on the overall ability of peers to reach their desired service levels. We also believe that this model is representative of several other cooperation scenarios occurring in peer-to-peer systems, such as cooperative query processing. The process by which nodes operate can be viewed as the dynamic construction of coalitions of nodes, cooperatively working to collectively achieve their service levels. A coalition that can keep all its members satisfied (i.e. attain their service levels) will persists since nodes do not change their behavior if they are satisfied. The simulation model is currently under developments. This work is part of a broader scope research which will be performed in second half of the project, and will include research on game theoretical approaches, auction-based mechanisms etc. applied to the problem of sharing of several resource types, including bandwidth, CPU, storage, as well as dynamic formation of prices.

References


