# P2P-DIET: An Extensible P2P Service that Unifies Ad-hoc and Continuous Querying in Super-Peer Networks *

Stratos Idreos and Manolis Koubarakis and Christos Tryfonopoulos

Intelligent Systems Laboratory
Dept. of Electronic and Computer Engineering
Technical University of Crete
GR73100 Chania, Crete, Greece
{sidraios, manolis, trifon}@intelligence.tuc.gr

## 1. INTRODUCTION

The main application scenario considered in recent peer-to-peer (P2P) data sharing systems is that of *ad-hoc querying*: a user poses a query (e.g., "I want music by Moby") and the system returns a list of pointers to matching files owned by various peers in the network. Then, the user can go ahead and download files of interest. The complementary *publish/subscribe* (pub/sub) scenario has started receiving attention only recently [1, 7, 8, 10]. In a pub/sub, a user posts a *continuous query* or *profile* to the system to receive notifications whenever certain *resources* of interest are *published* (e.g., when a song of Moby becomes available). Pub/sub can be as useful as ad-hoc querying in many target applications of P2P networks ranging from file sharing, to more advanced applications such as alert systems for digital libraries, e-commerce networks etc.

In this demo paper we present the system P2P-DIET, a service that unifies ad-hoc and continuous query processing in P2P networks with super-peers. Conceptually, P2P-DIET is a direct descendant of DIAS, a Distributed Information Alert System for digital libraries, that was presented in [7] but was not fully implemented. P2P-DIET combines ad-hoc querying as found in other super-peer networks [2] and SDI as proposed in DIAS. P2P-DIET has been implemented on top of the open source DIET Agents Platform (`http://diet-agents.sourceforge.net/`) and it is currently available at `http://www.intelligence.tuc.gr/p2pdiet`.

## 2. ARCHITECTURE

There are two kinds of nodes in P2P-DIET: *super-peers* and *client-peers* (see Figure 1). All super-peers are equal and

*This work was supported in part by the European Commission projects DIET (5th Framework Programme IST/FET) and Evergrow (6th Framework Programme IST/FET).


*SIGMOD 2004* June 13-18, 2004, Paris, France.


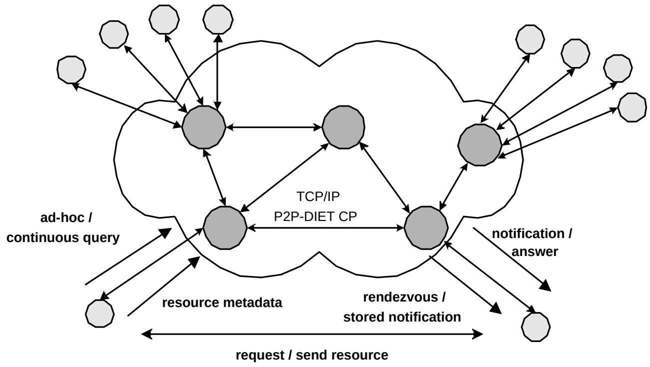


**Figure 1: P2P-DIET architecture**

have the same responsibilities, thus the super-peer subnetwork is a *pure* P2P network (it can be an arbitrary undirected graph).

The implementation of P2P-DIET to be demonstrated supports the Information Retrieval (IR) based model $\mathcal{AWP}$ for annotating resources and posing queries [7]. $\mathcal{AWP}$ is based on *attributes* with values of type *text*(interpreted under the Boolean model of IR) and supports *word proximity* operators [3]. As an example, the $\mathcal{AWP}$ query

$$AUTHOR = \text{“}\textit{John Smith}\text{”} \wedge TITLE \sqsupseteq (\textit{peer-to-peer} \wedge (\textit{selective} \prec_{[0,0]} \textit{dissemination} \prec_{[0,3]} information))$$

requests all resources (presumably papers) that have John Smith as their author, and their title contains the word "peer-to-peer" and a word pattern where the word "selective" is followed immediately by the word "dissemination" which, after at most 3 words, is followed by the word "information".

Ad-hoc/continuous query processing and notification propagation is implemented by broadcasting messages (using shortest path trees) in the super-peer backbone. In addition, the following data structures are exploited at each super-peer: (i) a *resource index* (an inverted file) for answering queries about local resources, (ii) a *query index* that is used by algorithm BestFitTrie [6] to find efficiently which continuous queries are matched by an incoming notification, and (iii) a *poset* that keeps track of subsumption relations among the continuous queries posted to a super-peer by its clients or forwarded by other super-peers (as in SIENA [1]).

Each super-peer serves a fraction of the client-peers and keeps *indices* on the resources of those client-peers. Resources (e.g., files in a file-sharing application) are kept at client-peers and they can only be requested from their owners (except in a special cases to be explained below). Client-peers are equal to each other since the software running at each client-peer is equivalent in functionality. A client-peer can request a resource directly from the resource owner client-peer. A client-peer is connected to the network through a single super-peer node, which is the *access point* of the client-peer. It is not necessary for a client-peer to be connected to the same access point continuously since client migration is supported in P2P-DIET. Client-peers can connect, disconnect or even leave the system silently at any time. To enable a higher degree of decentralization and dynamicity, we also allow client-peers to use *dynamic IP addresses*.

P2P-DIET offers the ability to add or remove super-peers. Additionally, it supports a simple fault-tolerance protocol based on *are-you-alive* messages. Finally, P2P-DIET provides message authentication and message encryption using PGP technology. For the detailed protocols of the system see [5].

In current work P2P-DIET is extended to support resource annotation by RDF and querying by RDF-based query languages [4]. Additionally, the super-peer backbone is implemented using the hypercube topology HyperCup [9] which allows for efficient broadcasting operations.

## 3. FEATURES TO BE DEMONSTRATED

Let us now describe a complete demonstration scenario of our system. Initially the super-peer backbone is created. The routing paths are reestablished whenever the interconnection topology of the super-peers is updated. Then, client-peers are attached to the network. The fault tolerance modules will allow the system to recover as super-peers or a client-peers fail. A client-peer may publish a resource by creating metadata for the resource and forwarding those metadata to its access-point. In this way, other users may see and request the resource. A client-peer may pose an one-time query to the system (by forwarding it to the access-point) to search for files on the whole network. The system will immediately reply with pointers (identifiers of remote client-peers and resources) to matching resources. The complementary scenario is that of a client-peer that subscribes (to its access-point) with a continuous query. Such a query will be broadcasted to the super-peer backbone and replicated at each super-peer. The system will reply with notifications (identifiers of remote client-peers and resources) whenever matching resources are published. The continuous query poset prunes broadcast messages. In the case that a client-peer is off-line at the time that a notification is produced due to one of its continuous queries, then the notification is stored (for a period of time that is a system parameter) to its access point and the client-peer is notified upon reconnection even if it migrates to a new super-peer. Moreover, a client-peer may arrange a rendezvous with a resource if the resource owner client-peer is off-line (the resource is unreachable). The previous access-point of the resource owner is responsible to notify this client-peer when it reconnects (even if it migrates to another super-peer) so as to upload the resource to the access point of the client-peer that requested it. Finally, a client-peer can use dynamic IP addresses and different access-point nodes of the network, request files directly from the resource owner client-peer and locate remote client-peers.

## 4. REFERENCES


[1] A. Carzaniga, D.S. Rosenblum and A.L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, August 2001.

[2] B. Yang and H. Garcia-Molina. Designing a super-peer network. In *Proceedings of the 19th International Conference on Data Engineering*, March 5–8 2003.

[3] C.-C. K. Chang, H. Garcia-Molina, and A. Paepcke. Predicate Rewriting for Translating Boolean Queries in a Heterogeneous Information System. *ACM Transactions on Information Systems*, 17(1):1–39.

[4] P. A. Chirita, S. Idreos, M. Koubarakis, and W. Nejdl. Publish/subscribe for rdf-based p2p networks. In *Proceedings of the 1st European Semantic Web Symposium*, May 2004.

[5] S. Idreos and M. Koubarakis. P2P-DIET: A Query and Notification Service Based on Mobile Agents for Rapid Implementation of P2P Applications. Technical Report TR-ISL-2003-01, Intelligent Systems Laboratory, Dept. of Electronic and Computer Engineering, Technical University of Crete, June 2003.

[6] S. Idreos, C. Tryfonopoulos, M. Koubarakis, and Y. Drougas. Query processing in super-peer networks with languages based on information retrieval: the p2p-diet approach. In *Proceedings of the 1st International Workshop on Peer-to-peer computing and Databases*, March 2004.

[7] M. Koubarakis, T. Koutris, P. Raftopoulou, and C. Tryfonopoulos. Information Alert in Distributed Digital Libraries: The Models, Languages and Architecture of DIAS. In *Proceedings of ECDL '02*, September 2002.

[8] P.R. Pietzuch and J. Bacon. Peer-to-Peer Overlay Broker Networks in an Event-Based Middleware. In *Proceedings of the 2nd International Workshop on Distributed Event-Based Systems*, June 2003.

[9] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. HyperCuP – hypercubes, ontologies and efficient search on peer-to-peer networks. In *Proceedings of the 1st Workshop on Agents and P2P Computing, Bologna*, 2002.

[10] W.W. Terpstra, S. Behnel, L. Fiege, A. Zeidler, and A.P. Buchmann. A Peer-to-Peer Approach to Content-Based Publish/Subscribe. In *Proceedings of the 2nd International Workshop on Distributed Event-Based Systems*, June 2003.