A SIMULATOR FOR A MOBILE PEER-TO-PEER DATABASE ENVIRONMENT

Verena Kantere, Konstantina Palla, Kostas Patroumpas, Timos Sellis

School of Electr. and Comp. Engineering
National Technical University of Athens
Heroon Polytechniou 9, 15780 Zografou, Greece
{verena, kpalla, kpatro, timos}@dbnet.ece.ntua.gr

ABSTRACT

We present a simulation environment that can be employed to study P2P mobile networks that are fast-evolving in both their topology and their content. This simulator implements a proposed P2P architecture based on Mobile Agent and Active Database technology and can be employed in order to build simulated mobile networks that are characterized by a diversity in peer needs, specifications and capabilities.

1. INTRODUCTION

We are interested in the special kind of peer-to-peer (hereafter P2P) systems that are fast-evolving in both their network topology and the content of the peers. Specifically, we focus on pure such systems that comprise mobile peers and seek and accumulate location-based information. A challenging issue for such systems is to design an appropriate mechanism for processing queries. Since both the data content of the peers as well as their acquaintances, change rapidly the typical P2P querying techniques become inappropriate.

Our motivating example is the application of location-based services on a P2P network formed by the cars in a city. The peers (cars) form a wireless ad hoc network. Each peer can communicate with others within the communication radius. Peers gather traffic information about the area that they move or stand and express queries looking for traffic information in other areas. Peers are not responsible for routing the queries in the system. Instead, each query is responsible for its own routing. Peers contain necessary geographic information. The queries can use this information in order to implement their routing algorithm according to their specialized routing demands. Hence, there is no global routing policy in the P2P network.

In [1] we have proposed a solution for answering queries in pure fast-evolving P2P systems based on mobile agents implemented with Event-Condition-Action (hereafter ECA) rules and employing existing active database technology. We have argued that using a mobile agent framework in this context would lighten the peer infrastructure needed for instant and continuous query processing, would alleviate the necessity for a sophisticated policy of disseminating information-events and queries and would allow an adaptable query management according to personalized specifications of peers. The proposed system node architecture employs active database technology in order to cope with the scalability and performance issues that are very critical for the feasibility of a fast evolving P2P system. The main advantages of implementing the agents using ECA rules is that they assure a secure way of extracting information from an autonomous database and that they do not demand a specialized platform to run on. Rather than that, an aDBMS with basic functionalities is adequate. The proposed architecture respects the self-manageable and diverse nature of mobile agents while it guarantees their easy migration from peer to peer and the controllable but fast deployment on them. Furthermore, an agent can be adaptable to an assortment of heterogeneous peers, provided user-defined procedures and management policies. We have argued that the proposed architecture is efficient and scalable and thus ideal for very dynamic systems that lack any kind of centralized or distributed control such as fast-evolving pure P2P networks.

Demo Proposal

We have implemented the architecture for fast evolving P2P networks proposed in [1] as an extension of the Network Simulator (hereafter NS-2) [2]. This extended simulation environment can be employed in order to build simulated mobile networks that are characterized by a diversity in peer needs, specifications and capabilities. Specifically, this environment can simulate mobile networks that implement various routing algorithms that change their policy dynamically according to the network situation and peer requirements; moreover, the employment of mobile agents enables the simulation of various query expressions and query management methods as well as multi-query optimization. Finally, the extended simulator can give the opportunity to deal with additional issues, such as distributed caching and fault tolerance.
In this demonstration we will present our extended network simulator in order to make apparent its enhanced capabilities and possibilities for implementation and experimentation on the above network issues.

2. A MOBILE PEER DATABASE

Very briefly, we summarize the coarse architecture of a P2P system that supports mobile agents based on Active Database technology. The proposed peer architecture is presented in Figure 1. Each peer consists of (a) the Peer aDBMS, where data are stored and active rules are evaluated, (b) one or more Client modules, which provide and consume information to/from the aDBMS, (c) a P2P Interface through which the user communicates with the P2P Layer and the aDBMS, and (d) the P2P Layer. The latter comprises (i) the Acquaintance Manager (AcqM), which is responsible for managing acquaintances, (ii) the Mobile Agent Manager (MAM), which is responsible for handling the the mobile agents. For more information on the mobile peer architecture as well as the architecture of the mobile agents, the reader is referred to [1].

3. SIMULATOR FOR PEER MOBILE DATABASES

In order implement the above mechanism, we have employed the widely known and used simulation environment NS-2 [2] and the library of the mobile database SQLite [3].

Network Simulator

NS-2 is an object-oriented simulator for networks implemented in C++ and oTCL [4]. It is an open-source tool that has been widely employed in the international computer science community. NS-2 is a powerful multi-purpose simulation environment that is suitable for both wireless, mobile and satellite networks.

Simulator Extension

We have enhanced the basic architecture of the mobile wire-lesss node of NS-2 by creating the following new objects:

- **mobDB**: an autonomous database that can be managed and accessed locally,
- **MAM**: a module that is responsible for the management of incoming and outgoing mobile agents,
- **AcqM**: a module that receives and transmits packets with encapsulated mobile agents,
- **MobAgent**: implements the mobile agent that encapsulates the query and any relevant meta-data (i.e. for query management and routing).

In case of an outgoing query, the object MAM serializes the ECA rules and encapsulates them into the network packet. This packet is propagated to AcqM which sends it in the network. In case of an incoming query, AcqM de-serializes the network packet and propagates it to MAM which creates the local instance of the respective mobile agent, MobAgent. Furthermore, it continues with the execution of the ECA rules in the agent. These rules are parameterized and are instantiated with appropriate parameter values.

In order to satisfy the requirement for a local active database we have employed the library of SQLite [3]. The latter, implements a database with a very small footprint. This database is actually a SQL engine that can write/read to/from files without the requirement of a central server. Each extended NS-2 node disposes a mobDB object that realizes the communication with the local SQLite.

4. DEMONSTRATION

In this demonstration we present the extended NS-2 simulator summarized above. The demonstration focuses on examples of our motivating example, i.e. the application of location-based services in a wireless ad hoc network of peer-cars in a city road network. We employ a visualization tool on which we demonstrate this wireless network on a part of the road network of the city of Athens. The examples comprise the creation of continuous queries looking for traffic information in specific areas. The queries are encapsulated in respective mobile agents. We demonstrate how mobile agents are routed by hosting peers according to the routing algorithm that they specify; also, how they remain in the same area by hopping to appropriate peers. Moreover, we show how the agents are deployed and resumed on the hosting peers, by installing, evaluating and un-installing their ECA rules on the local database.

5. REFERENCES


224