Demo of DANTE: A Self-Adaptable Unstructured Peer-to-Peer Network

Luis Rodero-Merino*, Luis López*, Antonio Fernández*, José Antonio Pérez*, Juan Francisco Morcillo*, Vicent Cholvi†
*LADyR
Universidad Rey Juan Carlos
Mostoles, Spain, 28933
Email: {lrodero, anto, llopez}@gsyc.escet.urjc.es, io3383@esdebian.org, juanfran83@gmail.com
†Universitat Jaume I
Castellón, Spain, 12071
Email: vcholvi@lsi.uji.es

***Abstract*— We propose a demo of DANTE[1], an unstructured peer-to-peer network with self-adapting capabilities. DANTE is able to adapt its topology to the traffic on the network. The adaptation mechanism drives the system to an optimal topology when the network is under very high or very low load conditions. We have developed both a simulator of DANTE and a graphical application that allows to see the network topology evolution. The goal of this demo is to introduce the fundamentals of DANTE, and use our software to show it at work.**

I. Introduction

Peer-to-peer (P2P) systems have obtained great popularity during the last few years. The basic idea of P2P systems is that all peers are, at the same time, clients and servers, that is, they offer resources to other nodes and use resources from other nodes. P2P systems represent a new and powerful paradigm that differs from the traditional client-server architecture where each participant has a specific role. This paradigm has brought the necessity for novel solutions to deal with some of its limitations. More specifically, the research community has devoted many efforts to develop new and efficient techniques for the location of resources. Traditional location solutions, like using a centralized directory (Napster) or flooding (Gnutella), have shown serious drawbacks: vulnerability, lack of scalability, etc. Trying to solve those problems, researchers have proposed new searching mechanisms, like random walks or structured networks.

Our system, DANTE is an unstructured system that combines resource searching by random walks with a self-adapting topology mechanism. Networks where nodes change their connections, and hence the overlay topology, to adapt to changes on network conditions (such as traffic load) are said to have a *Dynamic Adaptable Network Topology*. In DANTE, the goal is to make the network topology to evolve between a star-like and a random-like one as the load on the network varies. Previous results [1] show that these are the optimal configurations for very low and very high loads.

The topology adaptation mechanism proposed here is inspired on [2]. The adaptation is performed by the nodes, that periodically run a certain algorithm (the same for all of them) that dictates how they must change their connections. This algorithm uses local knowledge. Then, neither a central system nor global knowledge is needed, making DANTE suitable to be implemented in real world peer-to-peer networks.

DANTE is not the first system that combines random walks and dynamic topologies, there are previous proposals like Gia [3] or the work of Lv. *et al* [4]. Nonetheless, DANTE is the first system that uses Guimerà results about optimal topologies, and solves all the problems to apply them to P2P systems. Besides, it uses a overall simpler reconnection mechanism than those described in [3] and [4].

[1]From Dynamic Adaptable Network TopologiEs.

II. Dynamic Adaptable Network Overlays

It is well known that the performance of random walks is highly dependent on the overlay structure of the system [5], [6]. The approach traditionally taken in the literature to model this begins by assuming that nodes know their own resources plus the ones held by their immediate neighbors. In this case, if some peer becomes a central node (all participants are connected to it) it will know all the resources present in the whole system and will be able to correctly answer all queries. In a star-like topology a few nodes become central and all nodes in the system are connected only to them. Hence, all searches are solved with at most one hop.

With this argument we understand that, in a non congested scenario, the optimal topology is a highly polarized star-like structure, as is stated by Guimerà [1]. However, this situation is inefficient if congestion considerations become relevant, since the central node may become overloaded. This is supported by the results in [7], where it is shown that high-degree nodes (those having most connections) support most of the shared load. Guimerà shows that the optimal network topology is a homogeneous-isotropic one in the presence of severe congestion.

DANTE implements a reconnection technique that adapts the overlay network topology to the load on the system. The resulting topology tends to a star-like when congestion is small and to a random-like when congestion is large. For intermediate loads, some peers become hubs, that is, nodes that have much more connections than the average degree on the P2P system. Hubs have a wide knowledge about the network contents, but are not central (not all nodes are connected to them), for this reason, they allow solving queries in few hops without becoming as congested as central nodes.

III. DANTE Reconnection mechanism

In DANTE each node can establish connections to other nodes. We say that a connection is *native* for the establishing node and *foreign* for the accepting node. Nodes can change their native connections, but not their foreign ones.

The number of native connections of each node is fixed. Nodes periodically run a *reconnection mechanism* that, using a particular algorithm, changes this connections. The algorithm used to choose which peers each node connects to is based on an *attachment kernel* Π_i, which determines the probability of a particular node to be connected/rewired to node i. The proposed kernel has the form

$$\Pi_i \propto k_i^{\gamma_i} \tag{1}$$

where k_i denotes the number of links of node i and

$$\gamma_i = \begin{cases} 2 \text{ if } i \text{ is not congested} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

If some node receives more queries that it can process, we say that node is *collapsed* or *congested*. The algorithm tries to make well connected nodes more attractive, so connections will tend to point to those peers as they know more about the network. But if some node gets collapsed, then it stops being attractive, and neighbors will disconnect from it quickly.

The rationale behind Equations 1 and 2 is explained as follows. First, remark that we assume that each node knows its resources and the resources of its one-hop neighbors. We note that by taking a value of $\gamma_i = 0$ for all nodes, we obtain a random-like topology (intuitively, all nodes have the same probability of being chosen for a new connection [8]); in turn, if the value of γ_i is strictly greater than 1, we obtain a star-like (the more connections one node has, the more likely it will be chosen by other nodes, finally building a star-like topology [8]). Consequently, in [2] is established that the value of γ_i will be either 2 if the node is not collapsed and 0 otherwise. Thus, the network will evolve towards a random-like topology when the nodes become collapsed, or towards a star-like topology otherwise. It is easy to realize that the value of γ_i for not collapsed nodes has a strong impact on the way topology evolves.

To compute congestion, each node keeps track of the time to process queries received during the last period. Let μ_i be the processing rate of node i, and λ_i the number of queries processed by it during the last minute. Then we say that:

$$\text{if } \lambda_i \geq \mu_i \Rightarrow i \text{ is congested} \tag{3}$$

The algorithm implemented by the *reconnection mechanism* can be summarized in the following steps (assuming node has n *native* connections):

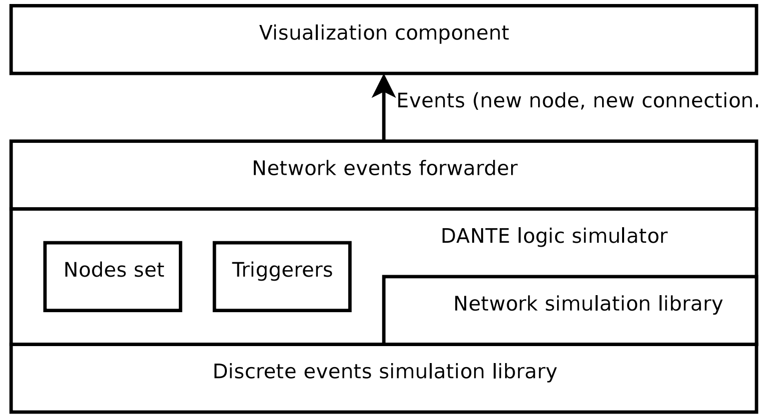

Fig. 1. Simulation software modules

1) Get list of candidates C.
2) For each candidate i in C, compute Π_i.
3) Assign to each candidate $i \in C$ a probability p_i, where p_i is computed as:

$$p_i = \frac{\Pi_i}{\sum_{j \in C} \Pi_j} \tag{4}$$

4) Choose n peers from list of candidates. Candidate i is chosen with probability p_i.
5) Disconnect present *native* connections and redirect them to the chosen candidates.

A. Candidates search

To gather the set of candidates required by the *reconnection mechanism* DANTE uses a special message that follows a random walk. When its TTL expires, the list of traversed peers is sent to the origin node.

IV. Simulation software

Our simulation software is implemented in Java. Its structure is depicted on Figure 1.

The main functionality is implemented by the *DANTE logic simulator* module. It contains the nodes in the system and the triggerers that start processes such as reconnections or searches for resources. It uses a *Network simulation library* that provides basic network concepts such as node and link. Both components use a *Discrete events simulation library*.

The *Network events forwarder* starts the simulation, and listens for network events such as the connection of two nodes, etc. It forwards the events to the *Visualization component*. This component shows the network topology in a graphical window, see Figures 2 and 3.

In Figure 2 we can see an example of an execution of DANTE, with high capacity nodes and low load on the network. The topology has evolved to a centered topology (starting from a random-like one). With other configurations DANTE evolves differently. Greater load and nodes of less capacity would make the network keep a random-like topology, like in Figure 3.

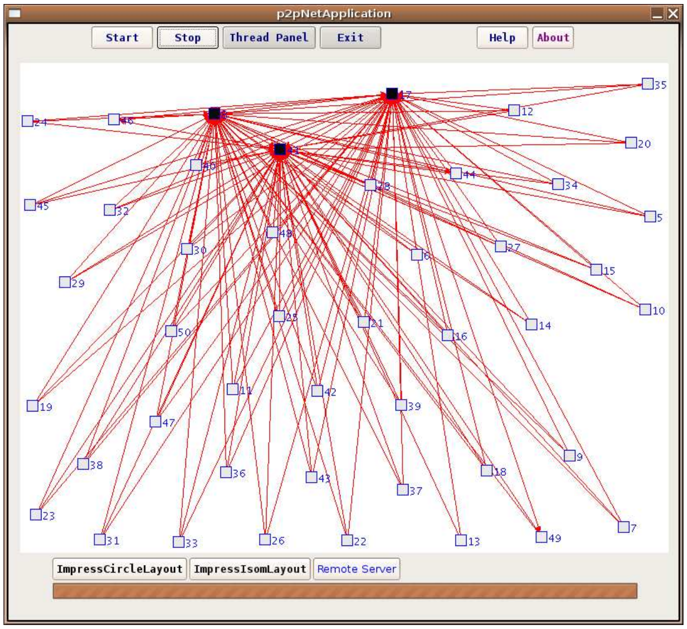

Fig. 2. Example of simulator execution, low load

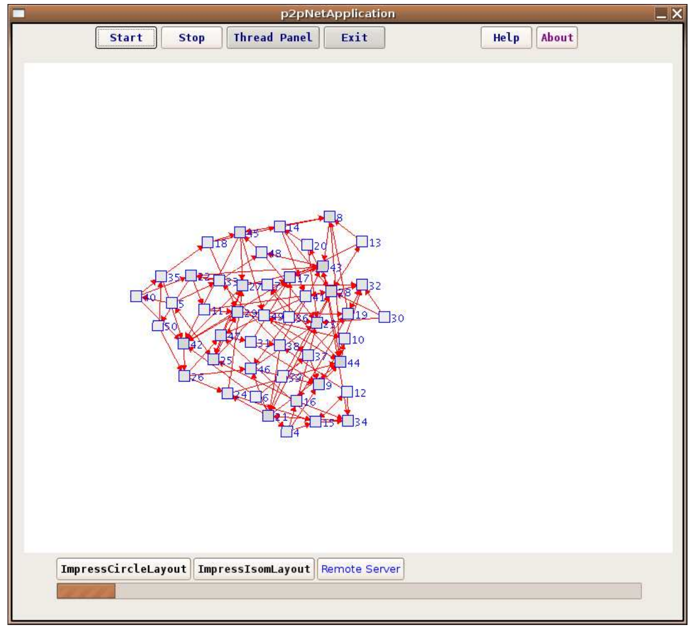

Fig. 3. Example of simulator execution, high load

A. Simulations parameters

Simulations are tuned using several parameters. The most important are the nodes capacities and the system load.

Each node i capacity is set by two values: *processing power* C_i and *bandwidth* BW_i. C_i is used to compute how much virtual time it takes to process each resource search, res/C_i, where res is the number of resources checked for that search. On the other hand, BW_i determines how much it takes to send a message. Both values are used to compute μ_i (see equation 3).

The system load is set by the *time between searchs* parameter. Each node periodically performs a new resource search, with a period that is computed as the inverse of that parameter. Resources to look for are chosen at random using an uniform distribution. Each resource is held by only one node.

V. Demo

In this demo, we first would make a brief introduction to DANTE's basic ideas, explaining the fundamentals of the reconnection mechanism. Then, we would use the software described above to run some executions of DANTE to show how DANTE topology adapts as intended.

VI. Conclusions

This demo aims to show DANTE ideas at work. DANTE nodes run a reconnection mechanism that makes the overlay topology evolve to an optimal configuration, without the need of global knowledge or a central coordinator. We think this in a novel approach that attendants can find interesting.

VII. Demo requirements

This demo does not need any special setup. Only space for one laptop and one monitor.

VIII. Acknowledgments

This work was partially supported by the Spanish Ministry of Science and Technology under Grant No. TSI2004-02940 and TIN2005-09198-C02-01, by Bancaixa under Grant No. P1-1B2003-37 and by the Comunidad de Madrid under grant S-0505/TIC/0285.

References

[1] R. Guimera, A. Diaz-Guilera, F. Vega-Redondo, A. Cabrales, and A. Arenas, "Optimal network topologies for local search with congestion," *Physical Review Letters*, vol. 89, November 2002.

[2] V. Cholvi, V. Laderas, L. López, and A. Fernández, "Self-adapting network topologies in congested scenarios," *Physical Review E*, vol. 71, no. 3, p. 035103, 2005.

[3] Y. Chawathe, S. Ratnasamy, N. Lanham, and S. Shenker, "Making gnutella-like p2p systems scalable," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM 2003)*, Karlsruhe, Germany, August 2003, pp. 407–418.

[4] Q. Lv, S. Ratnasamy, and S. Shenker, "Can heterogeneity make Gnutella scalable?" in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, Cambridge, United States, March 2002, pp. 94–103.

[5] L. A. Adamic, B. A. Huberman, R. M. Lukose, and A. R. Puniyani, "Search in power law networks," *Physical Review E*, vol. 64, pp. 46 135–46 143, October 2001.

[6] C. Gkantsidis, M. Mihail, and A. Saberi, "Random walks in peer-to-peer networks," in *Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004*, vol. 1, Hong Kong, March 2004, pp. 120–130.

[7] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of the 16th international conference on Supercomputing*, New York, New York, United States, June 2005, pp. 84–95.

[8] P. L. Krapivsky, S. Redner, and F. Leyvraz, "Connectivity of growing random networks," *Physical Review Letters*, vol. 85, pp. 4629–4632, November 2000.