Since the late nineties, peer-to-peer (P2P) protocols have become increasingly popular. Traditionally, these systems have been used to implement widely distributed applications, such as file-sharing services, as they provide efficient support for the discovery and distribution of information.

Several different structured P2P protocols have been proposed for distributed networks. Structured, in this context, refers to the protocols ability to organize network links and data to provide specific guarantees and bounds on performances. Recently, P2P protocols have also been exploited for information diffusion and aggregation, including resource discovery and system monitoring and community-based information dissemination. These applications introduce new requirements to P2P protocols, since information freshness, rather than information precision, is their main focus.

Epidemic-based (also known as gossip-based) P2P protocols are unstructured communication approaches that disseminate information in a manner similar to the spread of viruses in a biological community. They are often used to solve problems that might be difficult to tackle in other ways owing to the complex structure and dimension of the network and the fast rate of information change.

These developments have fostered an increasing interest within the research community in the conception and design of novel epidemic protocols. A typical issue is the need to consider classical non-functional requirements, such as scale and performance as foundational aspects of protocol design. Indeed, these protocols fit networks comprising hundreds of thousands of nodes characterized by frequent changes in shared data and affected by considerable churn rates.

Since it is unfeasible to obtain access to thousands of machines worldwide, the performances and limits of epidemic protocols are normally studied through simulations. In this sense, well-designed simulators facilitate the development of new protocols allowing for the simulation of many nodes within limited computational units. Properly designed simulators also ease the deployment of the protocols on a real infrastructure with minimal disruption to the code.

In the last decade, several P2P simulators have been proposed. Essentially, they differ in the level of abstraction provided, the programming language used and flexibility in developing protocols. To ease the evaluation and comparison of protocols, these simulators
are often bundled with several well-known protocols. Unfortunately, only a few P2P simulators are specific to unstructured protocols and only a subset of them provide a bundle of epidemic-based protocols as baselines for testing.

We developed Epeerdemics with the aim of filling this void. Epeerdemics is an extension to Overlay Weaver [3], an overlay construction toolkit widely diffused in the P2P community, mainly used for developing structured protocols. Epeerdemics is specifically designed to ease the development and testing of epidemic-based protocols. Protocols can either be developed from scratch or by extending the protocols provided with Epeerdemics. To this end, Epeerdemics provides two of the most used epidemic-based peer sampling protocols: Cyclon and Vicinity. Cyclon realizes an inexpensive peer sampling to build an overlay characterized by a random-graph structure. Vicinity is a semantic peer-sampling based protocol built on top of Cyclon that realizes an overlay network in which the links express similarity between peers.

From an operative point of view, each epidemic protocol is based on two threads, one active and one passive. The active thread is activated by a timer. When activated it selects one or more peers to communicate with and sends them (a subset of) its own knowledge. These interactions awaken, in the selected peers, the respective passive threads. Each passive thread answers by sending back to the sender (a subset of) its knowledge.

The amount and kind of information exchanged as well as the information retention strategy is protocol dependent. Epeerdemics supports programmers in implementing these choices and strategies. To implement an epidemic protocol with Epeerdemics, a programmer has simply to extend two Java classes: One for implementing the selection strategy to decide which peers to communicate with and one to specify the information retention strategy, i.e., which information to store or discard.

Here at ISTI-CNR, Epeerdemics has been used successfully for developing several different epidemic-based protocols. These range from simple enhancements to existing protocols to the definition of completely new protocols, including GoDel [1] a protocol for building Delaunay overlays and a gossip-based overlay construction for large scale online games [2].

References

