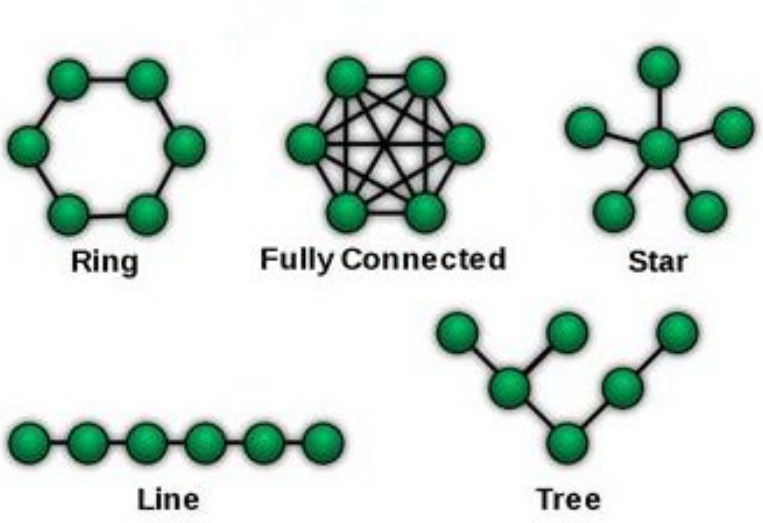
PINEAU Alexis
LORET Alexis

T-MAN: Gossip-based Overlay Topology Management

Márk Jelasity and Ozalp Babaoglu

Introduction

In large, dynamic, fully distributed system the way the nodes of the network are connected affects the functions and services that are implemented through the network. The way the nodes are connected is called the topology, it is the shape of the network. In a distributed network it is often important to keep a specific topology, as nodes can join or leave the network, the topology constantly changes, it can be hard to keep the right topology. The purpose of this paper is to propose a protocol that can manage such networks to get desired topologies, the name of this protocol is T-MAN.



Examples of topologies

The problem

From a given network (that can be random) we want to construct a topology. All nodes have to be connected to the right neighbour to get the right “network shape”. The input of the problem is a set of N nodes. Each node maintain a view, which is a set of c node descriptors (c is the number of links we want between each node in the target topology, the

descriptors contain address and properties (that can be ID, location, etc) of the node's neighbours. There is also a ranking function that can order the nodes of view.

The goal of T-MAN is to set the nodes of each node's views so that it matches the target topology. With a given node and its view the ranking function is able to get the c nodes that are the closest to the target topology. One way to get a ranking function is to define a distance function that defines a metric space over the set of nodes. The shorter the distance will be, the higher the ranking will be. However there are topologies that can't be defined by distance-based functions.

The solution

Protocol

The solution proposed by T-MAN is that every node execute a protocol in which they exchange their views with their closest neighbour through buffers and they select the c closest nodes according to the ranking function. This protocol is executed periodically so that in every cycle every node improve its view and the current topology becomes closer to the target topology.

The solution is based on two threads, an active one and a passive one. The active thread initiates communication with other nodes and the passive one waits for incoming messages.

When exchanging their view, nodes also add a random sample of the nodes from the entire network. This random sample is important when a node has a low quality neighbour set in large topologies. It reduces the number of exchanges it has to do before getting the right set of neighbours.

Self-healing

When nodes are constantly leaving or joining the network, it affects the quality of the topology. The solution of the authors is based on the age of the nodes. Each descriptors in the views have a new age field. In each cycle a few old descriptors are removed from the view. With this solution, it is expected to decrease the number of "dead links" and thus improving the topology quality.
Experiments have shown that this solution works but if the number of descriptors that are removed is to high it will damage the network.

Conclusion

T-MAN is a good solution for managing large dynamic distributed system topologies. The protocole is quite simple to understand and thus to implement and debug. T-MAN is useful because it can be implemented before the final topology is known. The ranking function can be generated dynamically so the topology can be created on the fly.

The authors are currently working on applications of T-MAN for jump-starting existing DHT implementations (like Kademlia) and providing them robustness in the presence of massive failures. They also want to better understand T-MAN behaviour and scope of performance. Finally, they want to study optimals ranking functions for given problems.