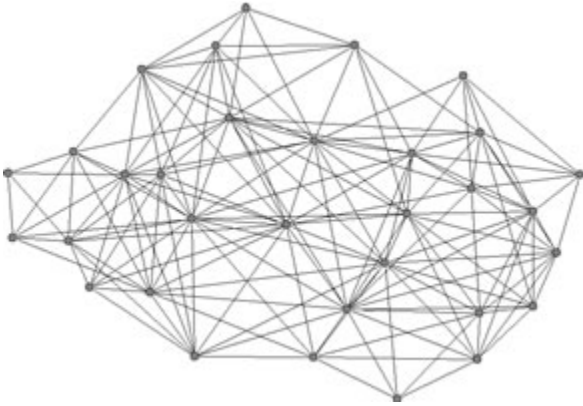


# Can

## Description

The Content-Addressable Network (CAN) is a framework for structured P2P systems based on a virtual d-dimensional Cartesian coordinate space on a d-torus. Nodes in a CAN graph have as an attribute the coordinates of a subspace of this space that are used in adding nodes and edges to the graph. Initially, the graph consists of one node and no edges. This initial node is assigned the entire virtual space. As nodes are added to the graph, they are assigned a subspace in the virtual space from a uniform distribution. So for a new node to join the CAN graph, it must find a node already in the graph and then using the routing mechanism, it should locate the node whose zone will be split. The system self-organizes to adjust to a new node by adding edges from the new node to adjacent nodes in the space. A typical CAN network graph is given below.



## Pros & Cons

Applicable to network graphs only. Since placement of system resources at nodes is strictly controlled in a centralized fashion, network evolution has costly overhead. Centralized control limits network robustness and node autonomy.

CAN model is advantageous for building systems where controlled resource placement is a high priority, such as distributed file storage. CAN model provides sub-linear search mechanisms.

### Applications

These algorithms can be used to generate CAN structured P2P graphs. These graphs can then be visualized and analyzed using Pajek or any other network visualization program.

### Implementation Details

Pseudo code for building CAN network model:

1. Add first node to the graph.
2. Choose a random position in the virtual coordinate space for new\_node.
3. Add new\_node to the graph.
4. Find the old\_node that owns that position.
5. Split the region occupied by old\_node along the longer axis and assign one half to the new\_node.
6. Fix the edges in the graph for both the old\_node and new\_node.
7. nodesAdded = nodesAdded + 1.
8. Repeat Step 2 if nodesAdded < desiredNetworkSize.

### Usage Hints

In the IVC Software Framework, simply select Modeling > P2P > Structured > CAN to generate a network. Use Analysis > Search > CAN Search to analyze the network.

### Acknowledgements

This package was developed by George Fletcher and Hardik Sheth. Hardik Sheth integrated the code into the IVC Software Framework.

### References

- Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. Proc. ACM SIGCOMM 2001, pp. 161-172, August 2001

## Links

[Source Code](#)