



# Content-addressable network

---

The **content-addressable network** (**CAN**) is a distributed, decentralized P2P infrastructure that provides hash table functionality on an Internet-like scale. CAN was one of the original four distributed hash table proposals, introduced concurrently with Chord, Pastry, and Tapestry.

## Overview

---

Like other distributed hash tables, CAN is designed to be scalable, fault tolerant, and self-organizing. The architectural design is a virtual multi-dimensional Cartesian coordinate space, a type of overlay network, on a multi-torus. This  $n$ -dimensional coordinate space is a virtual logical address, completely independent of the physical location and physical connectivity of the nodes. Points within the space are identified with coordinates. The entire coordinate space is dynamically partitioned among all the nodes in the system such that every node possesses at least one distinct zone within the overall space.<sup>[1]</sup>

## Routing

---

A CAN node maintains a routing table that holds the IP address and virtual coordinate zone of each of its neighbors. A node routes a message towards a destination point in the coordinate space. The node first determines which neighboring zone is closest to the destination point, and then looks up that zone's node's IP address via the routing table.<sup>[1]</sup>

## Node joining

---

To join a CAN, a joining node must:

1. Find a node already in the overlay network.
2. Identify a zone that can be split
3. Update the routing tables of nodes neighboring the newly split zone.<sup>[1]</sup>

To find a node already in the overlay network, bootstrapping nodes may be used to inform the joining node of IP addresses of nodes currently in the overlay network.<sup>[1]</sup>

After the joining node receives an IP address of a node already in the CAN, it can attempt to identify a zone for itself. The joining node randomly picks a point in the coordinate space and sends a join request, directed to the random point, to one of the received IP addresses. The nodes already in the overlay network route the join request to the correct device via their zone-to-IP routing tables. Once the node managing the destination point's zone receives the join request, it may honor the join request by splitting its zone in half, allocating itself the first half, and allocating the joining node the second half. If it does not honor the join request, the joining node keeps picking random points in the coordinate space and sending join requests directed to these random points until it successfully joins the network.<sup>[1]</sup>

After the zone split and allocation is complete, the neighboring nodes are updated with the coordinates of the two new zones and the corresponding IP addresses. Routing tables are updated and updates are propagated across the network.<sup>[1]</sup>

## Node departing

---

To handle a node departing, the CAN must

1. identify a node is departing
2. have the departing node's zone merged or taken over by a neighboring node
3. update the routing tables across the network.<sup>[1]</sup>

Detecting a node's departure can be done, for instance, via heartbeat messages that periodically broadcast routing table information between neighbors. After a predetermined period of silence from a neighbor, that neighboring node is determined as failed and is considered a departing node.<sup>[1]</sup> Alternatively, a node that is willingly departing may broadcast such a notice to its neighbors.

After a departing node is identified, its zone must be either merged or taken over. First the departed node's zone is analyzed to determine whether a neighboring node's zone can merge with the departed node's zone to form a valid zone. For example, a zone in a 2D coordinate space must be either a square or rectangle and cannot be L-shaped. The validation test may cycle through all neighboring zones to determine if a successful merge can occur. If one of the potential merges is deemed a valid merge, the zones are then merged. If none of the potential merges are deemed valid, then the neighboring node with the smallest zone takes over control of the departing node's zone.<sup>[1]</sup> After a take-over, the take-over node may periodically attempt to merge its additionally controlled zones with respective neighboring zones.

If the merge is successful, routing tables of neighboring zones' nodes are updated to reflect the merge. The network will see the subsection of the overlay network as one, single zone after a merge and treat all routing processing with this mindset. To effectuate a take-over, the take-over node updates neighboring zones' nodes' routing tables, so that requests to either zone resolve to the take-over node. And, as such, the network still sees the subsection of the overlay network as two separate zones and treats all routing processing with this mindset.

## Developers

---

Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Shenker

## See also

---

- Chord
- Content-addressable storage
- Pastry

- Tapestry

## References

---

1. Ratnasamy; et al. (2001). "A Scalable Content-Addressable Network" (<http://www.eecs.berkeley.edu/~sylvia/papers/cans.pdf>) (PDF). In Proceedings of ACM SIGCOMM 2001. Retrieved 2013-05-20. {{cite journal}}: Cite journal requires |journal= (help)
- 

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Content-addressable\\_network&oldid=1163183395](https://en.wikipedia.org/w/index.php?title=Content-addressable_network&oldid=1163183395)"