

Distance Halving: Continuous Graphs

Matthias Ohst

September 30, 2008

Contents

1	Introduction	1
2	Continuous Graphs	1
2.1	The Distance Halving Graph	2
2.2	From Continuous Graphs to Discrete Graphs	2
3	Insertion of Peers and the Principle of Multiple Choice	3
3.1	The Principle of Multiple Choice	3
3.2	Two Lemmas Concerning the Principle of Multiple Choice	4
3.3	Insertion of Peers	4
4	Routing in the Distance Halving Network	4
4.1	Simple Algorithm	5
4.2	Congestion Optimized Algorithm	5
5	Conclusion	6
	Appendix	6

1 Introduction

VICEROY is a network with a constant degree and a logarithmic diameter. Such a network is called a *degree minimized network*. But Viceroy is relatively complex. In this article, an elegant and simple alternative will be introduced: the DISTANCE HALVING network.

In 2003 MONI NAOR and UDI WIEDER developed the Distance Halving network. Their goal was not only to develop a new peer-to-peer network but they put great emphasis on the principle of *continuous graphs*. They are actually used in the networks CAN and CHORD but Naor and Wieder formalized it first.

2 Continuous Graphs

A graph is a pair (V, E) , where V is the vertex set and $E \subseteq V \times V$ is the edge set. In a discrete graph, the vertex set V is finite, while in continuous graphs the vertex set V is infinite. This section introduces the Distance Halving graph as an example for a continuous graph and describes how one can get a discrete graph of it.

2.1 The Distance Halving Graph

The Distance Halving graph $G = (V, E)$ consists of the vertex set $V = [0, 1] \subseteq \mathbb{R}$ and the edge set $E \subseteq V \times V$ with four types of edges ($x \in [0, 1)$):

- Left edges: $(x, \frac{x}{2})$
- Right edges: $(x, \frac{1}{2} + \frac{x}{2})$
- Backward left edges: $(\frac{x}{2}, x)$
- Backward right edges: $(\frac{1}{2} + \frac{x}{2}, x)$

Consider two edges (x_1, y_1) and (x_2, y_2) . If both are left edges or both are right edges, then $|y_1 - y_2| = \frac{|x_1 - x_2|}{2}$. Because of this fact, this network is called DISTANCE HALVING. Conversely, if both are backward left edges or both are backward right edges, then $|y_1 - y_2| = 2|x_1 - x_2|$.

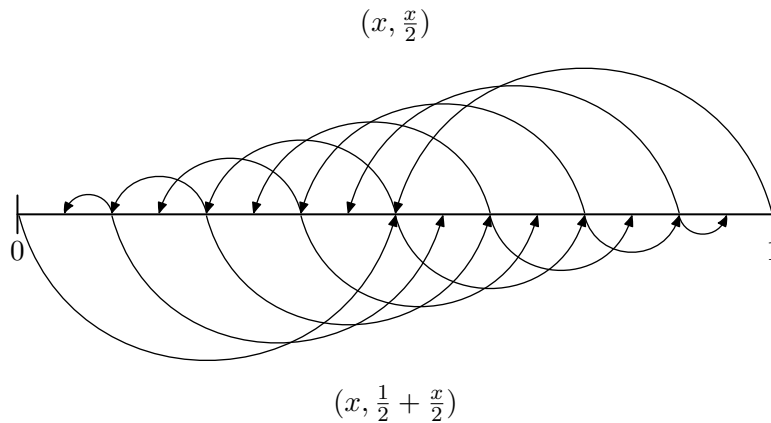


Figure 1: The Distance Halving graph.

2.2 From Continuous Graphs to Discrete Graphs

Continuous graphs cannot be used directly as network topology because of the infinite number of vertices. In order to get a discrete graph, the infinite vertex set V is partitioned into finite many intervals, which will be the vertices of the discrete graph and are called *segments*. In this case, the vertices or rather segments correspond to the peers in the network.

The simplest case would be to place the peers randomly in the interval $[0, 1)$. Then they are responsible for data from their position up to the position of their successor in the interval $[0, 1)$. Actually a modified positioning method is used in the Distance Halving network.

Denote the positions of the n peers by x_1, \dots, x_n in ascending order, i.e. $x_i < x_j$ for $i < j$. The peer x_i , $1 \leq i \leq n$, is assigned the segment $s(x_i) = [x_i, x_{i+1})$. There is an edge between two segments $s(x_i)$ and $s(x_j)$ iff points $u \in s(x_i)$ and $v \in s(x_j)$ exist such that (u, v) is an edge in the continuous graph. In addition there are edges between adjacent segments. So there is a ring structure. In this way, every path in the continuous graph can be mapped to a path in the discrete graph. Doing the discretization of the graph described above, one gets the DISTANCE HALVING network.

Because of the distance property, the degree of the Distance Halving network is constant if the ratio of the biggest to the smallest interval is constant. The edges of a segment map to an interval I which is for every type of edge at most twice as big as the segment itself. Let $\rho = \max_{1 \leq i, j \leq n} \frac{|s(x_i)|}{|s(x_j)|}$ be the ratio of the maximal segment size to the minimal segment size. Then the interval I can only intersect with at most $2\rho + 1$ segments. A constant ratio of $\rho = 4$ can be achieved by the *principle of multiple choice*, which will be presented in the following. Therefore the degree increases by a factor of nine because of the discretization, and hence the Distance Halving network has a constant degree.

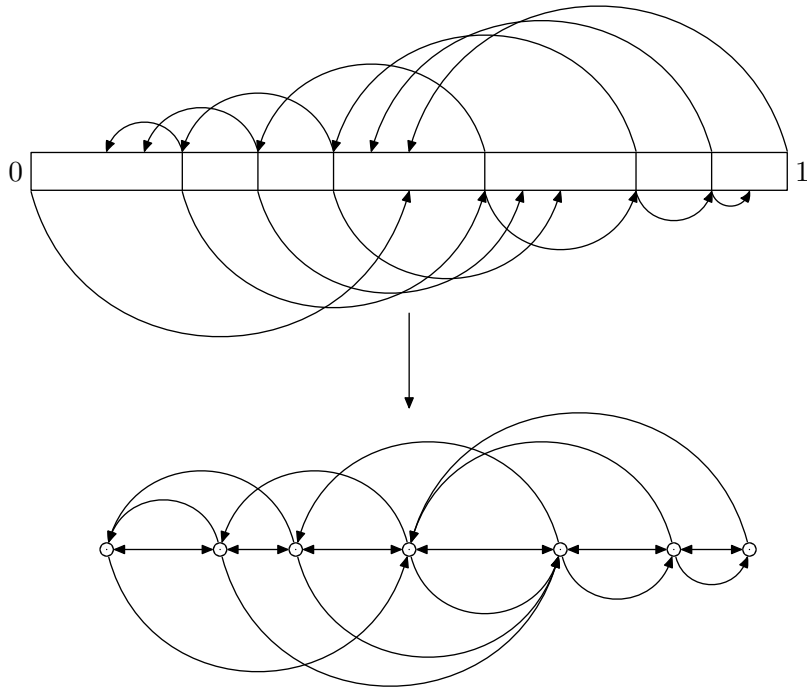


Figure 2: *Discretization of the Distance Halving graph.*

3 Insertion of Peers and the Principle of Multiple Choice

This section introduces the principle of multiple choice and proves that the degree of the Distance Halving network is constant if this principle is used.

3.1 The Principle of Multiple Choice

Instead of choosing a random position in the $[0, 1)$ ring during insertion, every peer looks first at $k = c \log n$ random positions $y_1, \dots, y_k \in [0, 1)$, where c is a suitable chosen constant. For every position y_i the size $a(y_i)$ of the segment $s(x_*)$ which surrounds the point y_i is measured, so the distance between the potential left and right neighbors in the $[0, 1)$ interval. The biggest of the segments found is chosen and the new peer is placed in the middle of that segment. In this way, always a relatively big segment is chosen, which implies that the distances between the peers are relatively uniformly.

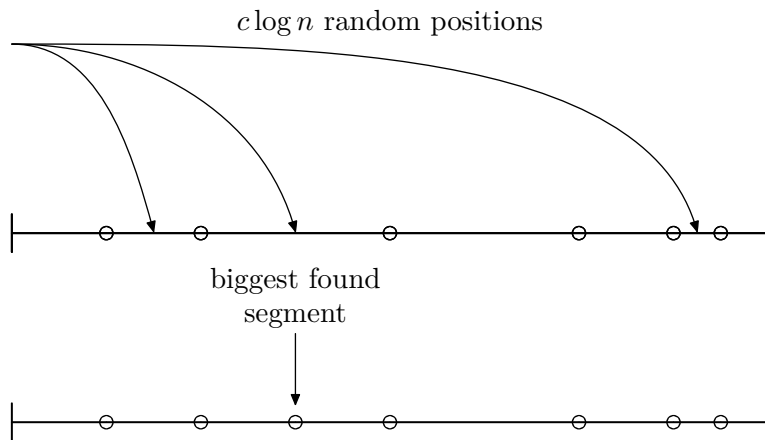


Figure 3: *The insertion of a peer in the Distance Halving network.*

3.2 Two Lemmas Concerning the Principle of Multiple Choice

Lemma 1. *If $n = 2^k$, $k \in \mathbb{N}$, peers are inserted in the $[0, 1)$ ring using the principle of multiple choice, only segments of sizes $\frac{1}{2n}$, $\frac{1}{n}$ and $\frac{2}{n}$ are left with high probability.*

Proof. Since the segments are divided in the middle, all segment sizes are powers of two. Hence it remains to show that no segments of size less than $\frac{1}{2n}$ and no segments of size greater than $\frac{2}{n}$ arise (with high probability). In order to show the second point, first another lemma is proven:

Lemma 2. *Let the biggest segment have the size $\frac{g}{n}$ (g may depend on n). Then after insertion of $\frac{2n}{g}$ peers all segments are smaller than $\frac{g}{2n}$.*

Proof. Consider a segment of size $\frac{g}{n}$. If $c \log n$ possible positions are examined during the insertion of every peer and $\frac{2n}{g}$ peers are inserted, the expected number of hits X in such an interval is

$$E[X] = \frac{g}{n} \cdot \frac{2n}{g} \cdot c \log n = 2c \log n.$$

Using the CHERNOFF bound (see theorem 4 in the appendix), one gets for $0 \leq \delta \leq 1$:

$$\Pr[X \leq (1 - \delta)E[X]] \leq n^{-\delta^2 c}.$$

If $\delta^2 c \geq 2$, all these intervals are hit at least $2(1 - \delta)c \log n$ times. Now one has to regard that every time an interval is divided by a peer the $(c \log n) - 1$ other hits of that peer (in possibly other big intervals) may not cause divisions. For $2(1 - \delta) \geq 1$, every interval of minimum length $\frac{g}{n}$ will be divided with high probability. \square

If one applies the previous lemma for $g = \frac{n}{2}, \frac{n}{4}, \dots, 4$, then no interval of size $\frac{g}{n}$ exists with high probability. The number of used peers is $4 + 8 + \dots + \frac{n}{4} + \frac{n}{2} \leq n$. After the last round there are no segments bigger than $\frac{2}{n}$. Since here only $\mathcal{O}(\log n)$ events have to arrive, the statement holds with high probability.

Now it remains to show that no segments smaller than $\frac{1}{2n}$ arise. The total length of all segments of size $\frac{1}{2n}$ is at most $\frac{n}{2}$ before insertion. The probability that only such segments are chosen by $c \log n$ tests is at most $2^{-c \log n} = n^{-c}$. For $c > 1$, a segment of size $\frac{1}{2n}$ is farther divided only with polynomially low probability. \square

3.3 Insertion of Peers

Until now we have disregarded that an approximation value of the number n of peers in the network is needed in order to check $c \log n$ positions on the ring during insertion. As in the Viceroy network, this estimation can be achieved by the distance of neighbors on the ring structure. By using the principle of multiple choice and particularly the previous lemma, the estimation in the Distance Halving network is exact except for a factor of 4, lastly the biggest segment has size $\frac{2}{n}$ and the smallest segment has size $\frac{1}{2n}$ with high probability.

During insertion the $c \log n$ segments that have to be checked are localized by a search. For this, $\mathcal{O}(\log n)$ steps are needed, as we will see shortly. After the biggest segment was chosen, the peer to insert will be embedded in the ring structure and then it establishes the other connections to the other peers with the help of the adjacent peers on the ring. Accordingly, the other neighbors in the network update, too.

4 Routing in the Distance Halving Network

We want to have a routing algorithm for the Distance Halving network which, in spite of the constant degree, only needs $\mathcal{O}(\log n)$ steps and at the same time distributes congestion uniformly. In order to show the basic idea of the routing in this network, first a simplified version will be presented which distributes congestion not as uniformly as the other algorithm does but also needs only a logarithmical number of steps.

4.1 Simple Algorithm

```

leftRouting(src, dest)
  if src and dest adjacent then
    send message from src to dest
  else
    newSrc  $\leftarrow$  leftPointer(src)
    newDest  $\leftarrow$  leftPointer(dest)
    send message from src to newSrc
    leftRouting(newSrc, newDest)
    send message from newDest to dest

```

Figure 4: Routing algorithm using only left edges.

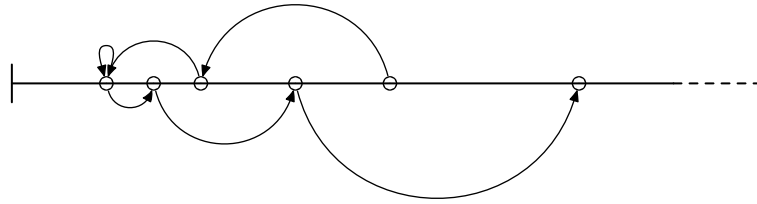


Figure 5: Example for routing in the Distance Halving network using only left edges.

This algorithm only uses the left edges. The source peer calculates two intermediate stations and reduces routing to half the distance. This continues until the source and destination nodes are adjacent. One could get the impression that the destination node takes part in the search, which is not correct. Lastly he does not “know” that he is in demand. The calculation of the intermediate stations is done by the source node. Then the intermediate stations must be told which path the message has to be carried on. Surely, routing also works using right edges:

```

rightRouting(src, dest)
  if src and dest adjacent then
    send message from src to dest
  else
    newSrc  $\leftarrow$  rightPointer(src)
    newDest  $\leftarrow$  rightPointer(dest)
    send message from src to newSrc
    rightRouting(newSrc, newDest)
    send message from newDest to dest

```

Figure 6: Routing algorithm using only right edges.

In both algorithms, the distance between source and destination is halved every recursion step, and every recursion step needs two steps. Since all interval sizes differ only by a factor of $\rho = 4$, the routing algorithm needs at most $1 + \log n$ recursions to deliver a message. It concludes that the routing cost is $2 \log n + 3$:

Lemma 3. *The routing in the Distance Halving network needs at most $2 \log n + 3$ messages and steps with high probability.*

4.2 Congestion Optimized Algorithm

Since the left and the right edges can be exchanged arbitrarily in these algorithms, the possibility arises to decide orientation (pairwise) by coin toss. While the first two algorithms tend to send

traffic into the outermost left or right corner, this algorithm arranges a good distribution of congestion. One can show here that congestion is very low [1].

```

randomRouting(src, dest)
  if src and dest adjacent then
    send message from src to dest
  else
    if coin shows number then
      newSrc  $\leftarrow$  leftPointer(src)
      newDest  $\leftarrow$  leftPointer(dest)
    else
      newSrc  $\leftarrow$  rightPointer(src)
      newDest  $\leftarrow$  rightPointer(dest)
    send message from src to newSrc
    randomRouting(newSrc, newDest)
    send message from newDest to dest

```

Figure 7: Routing algorithm using both left and right edges.

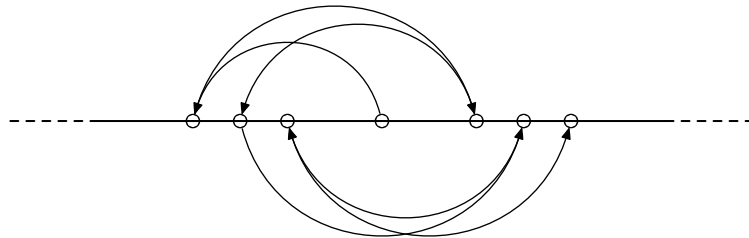


Figure 8: Example for routing in the Distance Halving network using both left and right edges.

5 Conclusion

We have seen that the Distance Halving network is a *degree minimized network*, i.e. it has a *constance degree* and a *logarithmic diameter*. This peer-to-peer network is an *elegant and simple alternative* to the complex *Butterfly* graph based *Viceroy* network.

Appendix

Theorem 4 (Chernoff bound). *Let X_1, \dots, X_n be independent Bernoulli experiments with probability $\Pr[X_i = 1] = p$ and $X = \sum_{i=1}^n X_i$. Then, for $\delta \geq 0$,*

$$\Pr[X \geq (1 + \delta)pn] \leq e^{-\frac{1}{3} \min\{\delta, \delta^2\}pn}.$$

Furthermore, if $0 \leq \delta \leq 1$,

$$\Pr[X \leq (1 - \delta)pn] \leq e^{-\frac{1}{2}\delta^2pn}.$$

References

- [1] MAHLMANN, Peter ; SCHINDELHAUER, Christian: *Peer-to-Peer-Netzwerke. Algorithmen und Methoden*. Berlin, Heidelberg : Springer-Verlag, 2007