Chapter 9

Mariposa: A Randomized Butterfly

In this Chapter, we describe Mariposa, a randomized routing network whose design philosophy differs from the randomized networks we studied in Chapters 7 and 8. The difference lies in whether a node learns about the random choices made by other nodes *before* or *after* link-establishment. We explain the point in more detail below.

In Randomized-Chord [ZGG03,GGG$^+$03], Randomized-Hypercube [CDHR03,GGG$^+$03], Symphony (see reference [MBR03] or Chapter 7), SkipNet [HJS$^+$03], skip-graphs [AS03] and Small-World Percolation Networks [MNW04], the routing network is constructed as follows. Each node generates some random bits locally, and establishes links with other nodes on the basis of the bits it generates. *After* link-establishment, a node can inspect the random bits of other nodes (typically, its neighbors with whom it has established links) for making good routing decisions. For example, "GREEDY with 1-LOOKAHEAD routing" (studied in Chapter 8) follows this paradigm. In Mariposa, each node first generates some random bits locally. It then inspects the random bits of a few other nodes *before* it establishes its long-distance links. The knowledge gained by inspecting other nodes' random bits is used for making good decision for link-establishment itself.

In a nutshell, all routing networks in Chapter 8 used the following sequence of operations:

- ➢ Generation of local random bits.
- ➢ Establishment of long-distance links.
- ➢ Inspection of non-local random bits for routing.

Mariposa uses the following sequence of operations:

- ➢ Generation of local random bits.
- ➢ Inspection of non-local random bits for establishment of long-distance links.
- ➢ Routing.

Mariposa is an interesting combination of butterfly networks and Kleinberg's small world construction [K00]. With $3\ell + 3$ out-going links per node, worst-case route lengths are $O(\log n/\log \ell)$ hops with high probability†, which is asymptotically optimal. Mariposa improves upon Viceroy [MNR02], an earlier butterfly-based construction that routes in $O(\log n)$ hops in expectation with $\Theta(1)$ links per node.

From a systems standpoint, Mariposa and Viceroy are complex constructions. However, understanding them yields insights into the design space of randomized routing networks, and sheds light on the algorithmic relationships between various randomized and deterministic routing networks.

Summary of Results

In §9.1, we describe the construction of Mariposa.

In §9.2, we prove the $O(\log n/\log \ell)$ bound for the worst-case routes in Mariposa.

In §9.4, we summarize and present directions for further research.

9.1 Mariposa: the Construction

Mariposa‡ is constructed over n nodes lying on the circumference of a circle. We will deal with two distributions:

- ⋆ Random Distribution: Each node has chosen its position independently and uniformly at random on the circumference of the circle.
- ⋆ Regular Distribution: The n nodes occupy positions corresponding to the corners of a regular n-gon circumscribed by the circle.

Each node in Mariposa maintains three real numbers:

---

†By "with high probability" (w.h.p.), we mean "with probability at least $1 - O(n^{-\lambda})$ for some constant $\lambda > 1$, for a system with n participants".

‡Mariposa means butterfly in Spanish.

✧ Position p

Each node will be assigned a position lying in the unit interval $\mathcal{I} = [0, 1)$. It is convenient to imagine $\mathcal{I}$ as a circle with unit perimeter. The binary operators $+$ and $-$ wrap around the interval $\mathcal{I}$. In other words, $x + y$ denotes the point that lies clockwise distance y away from x along the circle. Similarly, $x - y$ denotes the point that lies anti-clockwise distance y away from x.

For a random distribution of nodes, position p is chosen uniformly at random from $\mathcal{I}$. For a regular distribution of nodes, position p is a real number in $\mathcal{I}$, corresponding to one of the vertices of a regular n-gone circumscribed by the circle.

✧ Estimate $\tilde{n}$

For random distribution of nodes, in Chapter 4, we had described a distributed Network Size Estimation procedure by which a node can estimate the number of nodes as $\tilde{n}$, with the guarantee that $\tilde{n} \in [\frac{1}{4}n, 4n]$. Such an estimate can be made by inspecting the positions of a few adjacent nodes along the circle. We re-state this result as Lemma 9.2.1 in Section 9.2.

For regular distribution of nodes, $\tilde{n} = n$.

✧ Range r

A node chooses as its range r, a real number drawn from a *range probability distribution*

$$\mathcal{P}_{\tilde{n}} = 1/(x \ln \tilde{n}) \qquad \text{for } x \in [1/\tilde{n}, 1]$$

Distribution $\mathcal{P}_{\tilde{n}}$ is simply the continuous version of the discrete distribution in Kleinberg's paper [K00]. A node at position p with range r is said to *span* the interval $[p - r, p] \cup [p, p + r]$. Note that $[p - r, p]$ and $[p, p + r]$ could have more than one point in common if $r \geq 0.5$.

Hereafter, we will not make a distinction between random and regular distribution of nodes. We will assume that each node maintains $\tilde{n}$, an estimate of the total number of nodes. By setting $\tilde{n} = n$, we obtain the construction over a regular distribution of nodes.

Each node maintains $3\ell + 3$ outgoing links where $\ell \geq 1$. We will assume that $\ell = O(\text{polylog}(n))$. For $\ell \geq 2$, a node establishes 1 *short link*, 2 *intermediate links*, 2ℓ *long links* and at most ℓ *global links*. When $\ell = 1$, a node maintains 1 short link, 1 intermediate link, 2 long links and at most 2 global links. In any case, the total number of links is $3\ell + 3$ for

$\ell \geq 1$.

Short and Intermediate Links

A short link is established with the clockwise successors of a node. For $\ell \geq 2$, intermediate links are established with two nodes that are $\lceil \log \tilde{n} \rceil$ and $\lceil \log \tilde{n} / \log \ell \rceil$ hops away in the clockwise direction along the circle. When $\ell = 1$, only one intermediate link is established with the node that is $\lceil \log \tilde{n} \rceil$ hops away in the clockwise direction.

Short and intermediate links are used to route when the target is known to be nearby. Lemma 9.2.3 will show that a node that is $O(\log^2 n / \log \ell)$ hops away is reachable in only $O(\log n / \log \ell)$ steps.

Long Links

Long links lie at the heart of our protocol. A node at position p with range r partitions the interval $[p - r, p]$ into ℓ non-overlapping equi-sized sub-intervals and establishes one long link per sub-interval. It establishes ℓ additional links by partitioning the interval $[p, p + r]$ into ℓ non-overlapping equi-sized sub-intervals. Note that $[p - r, p]$ and $[p, p + r]$ would have more than one point in common if $r \geq 0.5$.

Let I_{sub} denote one of the sub-intervals of $[p - r, p]$ or $[p, p + r]$. By definition, its size is $|I_{sub}| = r/\ell$. Let p_{sub} denote the mid-point of I_{sub}. We now define an interval I_{search} with $|I_{search}| = 64 \ln^2 \tilde{n} / (\tilde{n} \ln \ell)$, centered at p_{sub}. Note that $|I_{search}|$ is independent of r. If $|I_{search}| \geq |I_{sub}|/2$, we say that I_{sub} is a *small* sub-interval. Otherwise I_{sub} is said to be a *large* sub-interval. If I_{sub} is small, we establish a link with the manager of the point $p_{sub} - r/(2\ell)$. If I_{sub} is large, we invoke a routine called SEARCH. The goal of SEARCH is to discover some node whose position is within I_{search} and whose range lies within the interval $[3r/(4\ell), 7r/(8\sqrt{\ell})]$. It is easy to see with $|I_{search}| < |I_{sub}|/2$, it is guaranteed that the span of such a node would include every point of I_{sub}. As Lemma 9.2.5, we will prove that w.h.p., all invocations of SEARCH succeed. This is because the interval I_{search} is sufficiently large in size.

For a node at position p with range r, we claim that all points within $[p-r, p] \cup [p, p+r]$ are reachable by short paths. To reach the manager of some point, we identify the sub-interval to which the point belongs and forward the lookup along that long link that corresponds to this sub-interval. If the sub-interval is small, we arrive at a node such that the destination is no more than $64 \ln^2 \tilde{n} / (\tilde{n} \ln \ell)$ away. At this point, intermediate and short links can carry out further routing. Lemmas 9.2.2 and 9.2.3 will show that this requires no more than $O(\ln n/\ln \ell)$ steps. If the sub-interval is large, we arrive at a node whose range is at most $7r/(8\sqrt{\ell})$. The idea is that shrinking by a factor of $7/(8\sqrt{\ell})$ limits the number of long links along any path to $O(\ln n/\ln \ell)$. We will prove our claims formally in Section 9.2.

One aspect of our construction remains. A lookup request can originate at a node that does not include the destination in its span. This might happen if $r < 0.5$. In such a case, how do we reach a node with range large enough to include the destination? Global links solve this problem.

Global Links

Global links are established by a node with range $r < 0.5$. Consider $I - [p-r, p+r]$ where I denotes the full circle. For $\ell \geq 2$, we partition the interval $I - [p-r, p+r]$ into ℓ equi-sized sub-intervals having size $(1-2r)/\ell$ each. For each sub-interval I_{sub}, we invoke SEARCH with the size and location of I_{search} being similar to our earlier description for long link establishment. The only change is that SEARCH looks for a node with range lying in the interval $[3(1-2r)/(4\ell), 1]$. When $\ell = 1$, we partition $I - [p-r, p+r]$ into two equi-sized sub-intervals with size $(1-2r)/2$ each. SEARCH is invoked twice to look for a pair of nodes, one in each sub-interval, with ranges lying in $[3(1-2r)/8, 1]$.

When a node initiates a lookup request, it forwards it along that global (or local) link whose span includes the destination point. Thereafter, the request is forwarded along a series of long links until we reach a sub-interval that is small. Hereafter, intermediate and short links are used for routing.

9.2 Analysis

Lemma 9.2.1. *With probability at least $1 - 2/n$, all nodes in a network of size n have $\tilde{n} \in [\frac{1}{4}n, 4n]$, assuming a random distribution of nodes.*

Proof. From Theorem 4.2 (for sufficiently small δ). □

We will establish that w.h.p., the worst case route length is $O(\ln n/\ln \ell)$ for $\ell = O(\text{polylog}(n))$. The overall proof idea is as follows. First, we show that small sub-intervals do not have high densities (A sub-interval is *small* if its size is less than $64\ln^2\tilde{n}/(\tilde{n}\ln \ell)$. In particular, we will show that w.h.p., no small sub-interval has more than $O(\ln^2 n/\ln \ell)$

nodes. Next, we will establish that with probability at least $1 - 3\ell/n$, all invocations of Search succeed. The resulting topology enjoys the property that path lengths of lookups are guaranteed to be as small as $O(\ln n/\ln \ell)$. Overall, we would have proved that w.h.p., the worst case route length for a lookup is $O(\ln n/\ln \ell)$.

Lemma 9.2.2. *With probability at least $1 - 2/n$, no small sub-interval has more than $O(\ln^2 n/\ln \ell)$ nodes.*

Proof. Using Chernoff Inequality and Lemma 9.2.1, we can show that with probability at least $1-2/n^2$, a particular sub-interval cannot be dense. Summing over all nodes, we obtain the requisite bound. □

The role of intermediate links is to route quickly to any node that is $O(\ln^2 n/\ln \ell)$ hops away along the circle.

Lemma 9.2.3. *Intermediate and short links can be followed to reach any node that is $O(\ln^2 n/\ln \ell)$ hops away in the clockwise direction in $O(\ln n/\ln \ell)$ steps.*

Proof. The longer of the two intermediate links can be followed in succession to reach a node that is at most $O(\ln n)$ hops away. This requires $O(\ln n/\ln \ell)$ steps. Then the shorter of the intermediate links can be followed to reach a node within $O(\ln n/\ln \ell)$ hops of the destination. This requires $O(\ln \ell)$ steps. Finally, $O(\ln n/\ln \ell)$ short links can be followed to reach the destination. Since $\ell = O(\text{polylog}(n))$, the total number of steps is $O(\ln n/\ln \ell)$. □

For small sub-intervals, long and global link establishment always succeeds. If the sub-interval is large, there is a chance that Search fails.

Lemma 9.2.4. *An invocation of* Search *fails with probability at most $1/n^2$.*

Proof. We will prove the lemma for long links. The proof for global links is along the same lines.

Search is invoked only if $|I_{sub}|/2 > |I_{search}|$. This implies $r/2\ell > 64\ln^2 \tilde{n}/(\tilde{n}\ln \ell)$, where $\tilde{n}$ is the estimate of the node that invoked Search. Thus, $3r/(4\ell) > 96\ln^2 \tilde{n}/(\tilde{n}\ln \ell)$ $> 16/\tilde{n}$ for large n. From Lemma 9.2.1, $16/\tilde{n} \geq 4/n$, which is definitely larger than $1/\tilde{n}$ for any node in I_{search} being probed.

When establishing long links, the goal of Search is to discover some node whose range lies in $[3r/(4\ell), 7r/(8\sqrt{\ell})]$. The probability that the range of a node with estimate $\tilde{n}$ lies in this interval is given by $p = \int_{3r/4\ell}^{7r/8\sqrt{\ell}} 1/(x \ln \tilde{n}) dx$. In the preceding paragraph, we showed that $3r/4\ell \geq 1/\tilde{n}$ for any node in I_{search}. Therefore, the value of the integral is $(\ln \frac{7}{6}\sqrt{\ell})/\ln \tilde{n}$. From Lemma 9.2.1, this quantity is at least $(\ln \frac{7}{6}\sqrt{\ell})/(2 \ln n)$.

$|I_{search}| = 64 \ln^2 \tilde{n}/(\tilde{n} \ln \ell)$. Lemma 9.2.1 yields $|I_{search}| \geq 8 \ln^2 n/(n \ln \ell)$ for large n.

Let us fix the position of the node which invoked Search. Consider the sequence of $n-1$ remaining nodes choosing their positions and ranges one by one. With probability $1 - |I_{search}|$, the position does not lie in I_{search}. Otherwise, with probability at least $(\ln \frac{7}{6}\sqrt{\ell})/(2 \ln n)$, Search succeeds. Thus the probability that no node makes Search succeed is at most $[1 - |I_{search}| + |I_{search}|(1 - \frac{\ln \frac{7}{6}\sqrt{\ell}}{2 \ln n})]^{n-1} \leq [1 - \frac{(8 \ln^2 n)(\ln \frac{7}{6}\sqrt{\ell})}{(n \ln \ell)(2 \ln n)}]^{n-1} \leq [1 - \frac{2 \ln n}{n}]^{n-1} = o(1/n^2)$. □

Lemma 9.2.5. *With probability at least $1 - 3\ell/n$, all invocations of* Search *succeed.*

Proof. Lemma 9.2.4 shows that the probability that a particular invocation of Search fails is at most $1/n^2$. Since there are at most $3\ell n$ total invocations, the probability that *any* of them fails is at most $3\ell/n$. □

The next lemma shows that although we allow Search to probe all nodes in a rather large sized I_{search}, the expected number of nodes it needs to probe is much smaller.

Lemma 9.2.6. Search *probes an average of $O(\ln n/\ln \ell)$ nodes before succeeding.*

Proof. In the proof of Lemma 9.2.5, we proved that the probability that a node in I_{search} makes Search invoked for long links succeed is at least $(\ln \frac{7}{6}\sqrt{\ell})/(2 \ln n)$. Thus the expected number of nodes to be probed before we succeed is at most $(2 \ln n)/(\ln \frac{7}{6}\sqrt{\ell}) = O(\ln n/\ln \ell)$. The same bound can be established for Search invoked for global links. □

We proved that w.h.p., all global and long links successfully get established. The resulting topology enjoys the property that all lookup paths are short.

Theorem 9.1. *With probability at least $1 - (6 + 3\ell)/n$, the worst case route length is $O(\ln n/\ln \ell)$ hops, when $\ell = O(\text{polylog}(n))$.*

Proof. From Lemmas 9.2.1, 9.2.2 and 9.2.5, we conclude that with probability at least $1 - (6 + 3\ell)/n$, all estimates of n are within a factor of four, no small sub-interval is dense and all long and global links get established. We show that the resulting graph has short diameter.

Routing proceeds in two phases. In the first phase, a lookup is forwarded along some long or global link whose range is guaranteed to contain the destination. The request then moves along a series of long links such that every node along the path has a range large enough to contain the destination in its span. The first phase starts at some node with range at most 1. From 9.2.1, when the first phase finishes, the last node will have range at least $1/4n$. Since each long link along the path shrinks the range by at least $\frac{8}{7}\sqrt{\ell}$, the first phase requires no more than $O(\ln n/\ln \ell)$ hops.

The second phase starts when we encounter a node with tiny range such that all its sub-intervals are small. At this point, the destination is only $O(\ln^2 n/(n \ln \ell))$ hops away (Lemma 9.2.2). Intermediate and small links can reach the destination in $O(\ln n/\ln \ell)$ steps (Lemma 9.2.3).

Total route length is thus $O(\ln n/\ln \ell)$ w.h.p. □

Reducing the Out-degree

We briefly outline a construction that requires only 4 links per node for $O(\ln n)$ average route length w.h.p. We set $\ell = 1$ and get rid of global links. Note that a fraction $\approx (c/\ln n)$ nodes will have their range smaller than c'/n for some constants c and c'. These nodes will not establish long links since their range is tiny. They will instead establish two global links each. Routing now requires that a lookup be forwarded to some node with tiny range. Hereafter, the usual protocol works. We can reduce the number of links to only 3 per node by removing the intermediate link as well. The resulting topology has an average route length of $O(\ln n/\ln \ell)$. However, the high probability bound no longer holds.

9.3 Intuition

In this Section, we develop intuition underlying the design of several routing networks: Chord, Kleinberg's construction [K00], Symphony [MBR03], sparse-Chord (studied in Chapter 8), Viceroy [MNR02], and Mariposa. We attempt to show that these networks constitute a continuum of design choices with Chord and Mariposa lying at two extremes.

Consider a cycle graph on n nodes where vertices are labeled $0, 1, 2, \ldots, n-1$ and there is an edge between node i and node $(i+1) \bmod n$. A message can be routed clockwise from a node to any other in at most $n-1$ steps. By the introduction of a few more links per node, routes can be made shorter.

Assume that a message destined for node $\mathbf{x_{dest}}$ is currently in possession of node $\mathbf{x_{src}}$. Let $d = (n + \mathbf{x_{dest}} - \mathbf{x_{src}}) \bmod n$, the distance between the nodes. Let h denote the number of 1's in $\mathbf{x_{dest}} \oplus \mathbf{x_{src}}$, the Hamming distance between the two nodes. With the exception of de Bruijn graphs, there seem to be two fundamental themes lying at the heart of existing routing protocols: A route diminishes either the distance d or the Hamming distance h to the destination. CAN [RFHK01], Chord [SMK$^+$01], Kleinberg's construction [K00], Symphony [MBR03], Viceroy [MNR02] and Mariposa [M03] are designed with d in mind. Hypercubes, Pastry [RD01a] and Tapestry [ZHS$^+$04] are designed with h in mind. Routes that diminish d do not necessarily diminish h and vice versa. However, the intuition behind both flavors of routing has commonalities, e.g., a protocol gradually diminishes the number of 1's in either d or h. We now present a unified picture of routing networks that diminish distance d during routing.

Distance Halving

Consider the function $\mathcal{C}_n(x) = (\ln nx)/\ln n$ for $x \in [\frac{1}{n}, 1]$. This is the cumulative probability distribution of $\mathcal{P}_n(x) = 1/(x \ln n)$ for $x \in [\frac{1}{n}, 1]$. For $x \in [\frac{1}{n}, 1]$, we will say that its *notch value* is $y = \mathcal{C}_n(x)$. While routing, let the current distance to the destination be $x_{current}$ with notch value $y_{current}$. Let $s = 1/\log_2 n$. If the current node has a link with notch value between $y_{current} - s$ and $y_{current}$, then we can forward the lookup along this link such that $x_{current}$ is at least halved and $y_{current}$ diminishes by at least s. The maximum number of times $x_{current}$ can be halved (and $y_{current}$ diminished by s) is at most $1/s = \log_2 n$. This intuition underlies several routing protocols that diminish distances.

1. The Chord routing network corresponds to every node establishing exactly $\log_2 n$ links corresponding to notch values $\langle 1-s, 1-2s, 1-3s, \ldots \rangle$. When a node wishes to route to a point $x_{current}$ away (with notch value $y_{current}$), it can immediately forward the lookup along a link such that $x_{current}$ is at least halved and $y_{current}$ diminished by at least $s = 1/\log_2 n$. The worst-case route length is thus $O(\log n)$.

2. In Kleinberg's construction [K00], each node establishes one long link with another node at a distance drawn from a discrete distribution which is quite similar to $\mathcal{P}_n$. This is equivalent to choosing a notch value uniformly at random from $[0, 1]$. Routing proceeds clockwise greedily. If the long link takes us beyond the destination, the request is forwarded to a node's successor. Otherwise, the long link is followed. Let

us denote the current distance to the destination by $x_{current}$ with notch value $y_{current}$. With probability $s = 1/\log_2 n$, the long link of the current node has notch value lying between $y_{current} - s$ and $y_{current}$. Thus the expected number of nodes that need to be visited before we arrive at a node which halves $x_{current}$ is $1/s = \log_2 n$. Effectively, in comparison with Chord, there is an inflation in average route lengths by a factor of $O(\log n)$. Kleinberg's routing scheme requires $O(\log^2 n)$ steps.

3. Symphony extends Kleinberg's idea in the following way. Instead of one long-distance per node, there are k long-distance links where $k \leq \log_2 n$. Effectively, a node gets to choose k notches uniformly from $[0, 1]$. Loosely speaking, when we are at $x_{current}$ (with notch value $y_{current}$), we need to examine roughly $(\log_2 n)/k$ nodes before we encounter some link that diminishes $x_{current}$ by at least half. Thus average route length for Symphony is $O(\frac{1}{k}\log^2 n)$ hops.

4. Sparse-Chord was studied in §8.5, and is defined as follows: Consider a cycle graph on n nodes. Let $b = \lceil \log_2 n \rceil$ bits. A node with ID $\mathbf{x}$ chooses an integer r uniformly at random from the set $\{1, 2, \ldots, b\}$ and establishes a link with node $\lceil \mathbf{x} + n/2^r \rceil \bmod n$.

   It is possible to route clockwise in $\Theta(\ln n \ln \ln n)$ steps in expectation by using a non-greedy decentralized routing strategy, as follows. Let the distance remaining to the destination be d. Let $b' = \lceil \log_2(4b \ln b) \rceil$ bits. If the long link of the current node corresponds to one of the top (most significant) $b - b'$ bit positions where d represented in binary has a 1, then forward the message along the long link. Otherwise, forward the message clockwise along the short link. Forwarding along a long link removes some 1 among the top $b - b'$ bits. The lower order b' bits act as a counter that diminishes by 1 whenever a short link is followed.

   The protocol is reminiscent of the classic coupon collection problem [MR95]. Essentially, we have to collect at most $b - b'$ coupons where the probability of collecting a coupon in one step is $1/b$. It is well known that all $b - b'$ bits can be collected in $2b \ln b$ steps in expectation. Building upon this intuition, it can be shown that on average, routing requires $O(b \ln b)$ hops. Since $b = O(\ln n)$, average latency is $O(\ln n \ln \ln n)$.

   With $\ell \leq \ln b$ links chosen uniformly out of the b possible, it can be shown that average latency diminishes to $O((\ln n \ln \ln n)/\ell)$. With $\ln \ln n$ links, average latency is only $O(\ln n)$. For large values of ℓ, a further improvement is possible. The key idea is that ℓ links can be used to fix $\lfloor \ln_2 \ell \rfloor$ bits in one hop. It can be shown that for large ℓ, routing requires $O((\ln n / \ln \ell) \ln(\ln n / \ln \ell))$ hops.

5. Viceroy: Before we describe Viceroy, let us investigate the Bit-Collection protocol further. Bit-Collection is only a factor $O(\log(\log n / \log \ell))$ more expensive than the best possible protocol. How could we possibly make it faster? By chaining the bits being collected. We illustrate the idea for a network with n nodes. Consider a node $\mathbf{x}$ with a finger that should point to $\lceil \mathbf{x} + n/2^r \rceil \bmod n$ for some integer r. This finger fixes the r^{th} most significant bit. If we could make it point to a node that fixes the $(r+1)^{th}$ bit, then we could hope to collect bits rapidly in succession. The key idea is to search for a pair of nodes, one each in the vicinity of $\mathbf{x}$ and $\mathbf{x} + n/2^r$ that both fix the $(r+1)^{th}$ bit. The two searches on average require only b steps each. How would routing work? If $\mathbf{x}$ wishes to send a message to some node, we first search for a node in the vicinity of $\mathbf{x}$ that fixes the top bit. This requires b steps on average. Then, routing proceeds rapidly by fixing successive top-order bits. A problem that emerges is that searches associated with the top order bits collectively introduce a bias of roughly $O(b^2)$. If every node maintains an additional pointer that points a fixed distance b away, the last stretch of length $O(b^2)$ can be covered in only $O(b)$ steps.

   The intuition developed in the previous paragraph is exactly how Viceroy [MNR02] would work if all nodes knew n precisely (that is, for a Regular distribution of IDs). Using the terminology of notches developed earlier in this Section, Viceroy assigns each node a notch value drawn uniformly at random from the set $\{1-s, 1-2s, 1-3s, \ldots\}$. The size of the set is $\log_2 n$. The relationship with Chord is the following. A Chord node uses the entire set for link establishment resulting in $\log_2 n$ links per node. However, a Viceroy node at position $p \in [0, 1)$ and notch value y (corresponding to distance $x = \mathcal{C}_n^{-1}(y)$), searches intervals centered around points p and $p + x$ for a pair of nodes with notch value $y - s$.

6. Mariposa goes a step further than Viceroy. Instead of being restricted to a finite number of discrete values for notch values, Mariposa chooses a notch-value in $[0, 1)$ uniformly at random. Moreover, Mariposa uses $3\ell + 3$ links per node as against $\Theta(1)$ links used by Viceroy.

9.4 Summary and Future Work

We described Mariposa, a combination of butterfly networks and Kleinberg's small world construction [K00] that routes in $O(\log n/\log \ell)$ hops with only $3\ell + 3$ links per node. The construction is an improvement over Viceroy [MNR02] which routes in $O(\log n)$ hops on average with only $\Theta(1)$ links per node. As future work, it would be interesting to understand further the relationships between randomized and deterministic routing networks.