Virtual Ring Routing Trends

Dahlia Malkhi[1], Siddhartha Sen[2], Kunal Talwar[1], Renato Werneck[1], and Udi Wieder[1]

[1] Microsoft Research Silicon Valley
{dalia,kunal,renatow,uwieder}@microsoft.com
[2] Princeton University
sssix@cs.princeton.edu

Abstract. Virtual Ring Routing (VRR) schemes were introduced in the context of wireless ad hoc networks and Internet anycast overlays. They build a network-routing layer using ideas from distributed hash table design, utilizing randomized virtual identities along a ring. This makes maintenance practical when nodes may enter or leave.
Previously, VRR was evaluated over a small wireless network and through medium-scale simulations, exhibiting remarkably good performance. In this paper, we provide a formal analysis of a family of VRR-like schemes. The analysis provides insight into a variety of issues, e.g., how well does VRR perform compared with brute force shortest paths routing? What properties of an underlying network topology make VRR work well?
Our analysis is backed by extensive simulation over a variety of topologies. Whereas previous works evaluated VRR over fairly small networks (up to 200 nodes), we are interested in scaling the simulations so as to exhibit asymptotic trends. Simulating network sizes beyond 2^{20} results in a memory explosion: In some of the topologies of interest, such as a 2-dimensional plane, the total memory taken up by routing tables is $\Omega(N^{3/2})$ for an N-node network. We devise a simulation strategy that builds necessary information on the fly using a Luby and Rackoff pseudo-random permutation, leading to simulations at a scale of 2^{32} nodes.

1 Introduction

Virtual Ring Routing (VRR) schemes were deployed for wireless ad hoc networks [4], for anycast Internet routing [5], and for scaling Ethernet [8]. Deviating drastically from any known method of compact routing [7], these practical systems borrow ideas from distributed hash table overlays, and use virtual addresses (aka flat labels) for routing. The vision behind these schemes is to have node identities that contain no structural information about the network. Hence, they support mobility naturally, and impose less administrative burden in assigning addresses. Additionally, they are easy to maintain, in that adding and removing nodes from the network is efficient, and incurs updates in only a small fraction of the nodes. In contrast to the well-founded theory of compact routing, there exists no rigorous analysis of VRR schemes. This paper tackles the formal analysis of a family of VRR schemes and provides insight into a variety of issues.

DHT overlays assign virtual identities (e.g., in the range [0..1], or integers) to nodes and maintain connections between nodes based on their virtual id's. When used for forming a network layer, DHT overlay techniques must be modified for the following reason. In an overlay network, a node p simply stores the name of each overlay neighbor q in a local overlay routing table; the lower-level networking layer facilitates the connection between p and q. However, in the absence of a network layer, it is not enough for p to remember q's name in order to connect to it.

VRR schemes such as [4, 5] resolve this issue by maintaining routing information between p and q along an entire physical path between them. This means that every node along a physical path from p to q has a routing table entry with the destination q in it, storing the next hop toward q. We note that other techniques that adapt DHT routing to the network layer exist, but are of no relevance here, e.g., write an entire path on the packet header at p [11], or route through landmark gateways [12, 10, 6].

To prevent confusion between routes at the different layers, we introduce some conventions.

Glossary: The entire node-by-node path determined by a routing scheme is called the *actual routing path*. It is induced by a sequence of hops, each hop between *virtual neighbors* in the virtual overlay. The physical path toward a virtual neighbor is carried along a *physical segment*, potentially composed of multiple nodes.

Routing efficiency is measured by its *stretch*: Given a pair of nodes, their routing stretch is the ratio between their actual routing path length and their shortest path length.

The overlay topology utilized in [4, 5] is a simple ring. Hence, overlay paths may take a linear number of virtual hops from a source to a destination. For example, say that we have a ring of nodes numbered [1..30]. The virtual ring route from node 5 to 15 goes through nodes $5, 6, ..., 15$ in succession. Each of these virtual hops is carried along a physical segment in the network. Thus, routing toward a virtual destination using overlay virtual hops could incur a linear stretch.

Fortunately, VRR allows *greedy hops* which considerably improve the routing efficiency. Imagine going along a physical path from node 5 to 6 in the above virtual path. Quite likely, this path crosses other physical paths, say 10 to 11, 20 to 21 and 28 to 29. When we reach the node en-route from 5 to 6 which is on the path from 10 to 11, VRR greedily chooses to route toward 11 instead of continuing toward 6.

Thus far, the advantage of using path intersection in greedy routing in this manner was evaluated over a small wireless network and through medium-scale simulations, exhibiting remarkably good performance.

1.1 Technical Approach

The high level intuition provided in [4] for constant expected stretch of VRR in a two-dimensional space with uniformly scattered nodes is as follows. The routing table at each node is populated with expected $O(\sqrt{N})$ randomly selected destinations. Hence, a greedy hop to the final target is expected after visiting $O(\sqrt{N})$ physical nodes. Unfortunately, this intuition is not easily turned into a rigorous analysis because of the subtle dependencies between the routing tables of neighboring nodes in the topology. Rather, our analysis builds from the fundamental probability of path intersection. For example, consider the Euclidean grid of dimension d. For reasonable selection of shortest paths between randomly chosen endpoints, intersection occurs with probability in $O(N^{-\frac{d-2}{d}})$. For the two-dimensional grid, this is constant. We call this the *intersection coefficient*, and denote it by p.

Two factors contribute to bound the routing path length. First, consider the last $2c/\sqrt{p}$ virtual identities preceding the target, for some constant c. They determine a collection of $c/\sqrt{p}$ physical segments with disjoint endpoints, that are hence independently assigned in the network. Any additional independently chosen segment intersects one of the segments in the collection with probability $1-(1-p)^{(c/\sqrt{p})}$. A collection of $c/\sqrt{p}$ additional, independently selected, segments intersects the first collection with probability $1-\left((1-p)^{(c/\sqrt{p})}\right)^{(c/\sqrt{p})} \approx 1-e^{-c^2}$, hence the expected collection size until intersection is in $O(1/\sqrt{p})$. Intuitively, this bounds the number of physical segments that are traversed to completion in the actual routing path to expected $O(1/\sqrt{p})$. A more precise argument, which considers the inter-dependencies among virtual hops in a routing path, is given in the body of the paper; it gives $O(\log N/\sqrt{p})$ expected number of completed physical segments.

So far, we have bounded the expected number of virtual hops that are made to completion in an actual routing path. We did not count the nodes in physical segments that are interrupted by greedy steps. Here, intuition suggests that a greedy step shortens the virtual distance to the target by an expected factor of two. However, we were unable to provide a formal proof for this property, due to the intricate dependency between the conditions imposed by a path traversed up to some point and the possible remaining virtual identity mappings.

Instead, we slightly modify the scheme to assist with the analysis. We modified the VRR scheme to allow a greedy hop only when, indeed, it reduces the virtual distance to the target by at least a constant factor α. Extensive simulation indicates that the modification has marginal (and even somewhat negative) effect on the actual routing complexity, e.g., for $\alpha = 2$. We can then bound the number of physical segments which terminate with a greedy hop by $O(\log_\alpha(N))$. Proving this for the original VRR scheme remains an open challenge.

1.2 Contribution

Leveraging the analysis we highlight above, we make the following contributions.

- Our analysis relates the path intersection coefficient p with an expected overall routing stretch of $O(\log N/\sqrt{p})$. We prove that this is tight up to a logarithmic factor, with a matching lower bound of $\Omega(1/\sqrt{p})$. Using this insight, one can predict the stretch of VRR schemes over any network topology, as the physical network topology determines path intersection probability p. For example, in a two-dimensional grid, two pairs of randomly selected endpoints have intersecting shortest-paths with constant probability. The expected stretch in the two dimensional grid is in $O(\log N)$.
  More generally, for the Euclidean grid of dimension d, intersection occurs with probability in $O(N^{-\frac{d-2}{d}})$, and the expected stretch is in $O(N^{\frac{d-2}{2d}} \log n)$.
- We readily determine the relationship between the overall routing table memory and the stretch. The network topology determines the expected number of overlay paths that pass through a certain node, and thus, the expected routing table size at a node. For example, in a d-dimensional grid, routing tables size is in $O(N^{1/d})$.
  Memory-stretch tradeoffs have been studied extensively in the theory of compact routing, and we can draw a comparison with VRR here. Methods were suggested that can achieve better characteristics: [1] gives a $O(k)$-stretch name-independent routing with $O(k^2 N^{1/k} \log^3 N)$ routing table resources per node for arbitrary graphs, and [2] gives a name-independent scheme for planar graphs with constant stretch and only polylogarithmic memory at each node. The advantage of VRR schemes is their simplicity and maintainability.
- We also extend our experiments to two overlay variants, using the same VRR methodology. One is a ring where each node has outgoing links to its k ring-successors, for a parameter k. The other is the ring with $k-1$ successors, and a k-th neighbor is selected from the virtual ring using a "small-world" distribution.

Our analysis is backed by extensive simulation over the two, three and four dimensional grids. Whereas previous works evaluated VRR over fairly small networks (up to 200 nodes), we are interested in scaling the simulations so as to exhibit asymptotic trends. However, directly simulating network sizes beyond 2^{20} results in a memory explosion: In some of the topologies of interest, such as a 2-dimensional plane, the total memory taken up by routing tables is $\Omega(N^{3/2})$ for an N-node network. Rather, we devise a simulation strategy that builds necessary information on the fly using a Luby-Rackoff pseudo-random permutation, leading to simulations at a scale of 2^{32} nodes.

2 Problem Description

We describe the VRR scheme in greater detail. The system is modeled as an undirected graph $G = (V, E)$. V is a set of $|V| = N$ nodes. Edges $(u, v) \in$

E indicate that u and v know each other, are physically connected and can communicate directly.

In VRR, every node v has a unique identifier $id(v)$ drawn uniformly at random from a range $R \gg N$ of integers. This defines a natural order on the identifiers and for the rest of the paper, we assume the identifiers simply define a permutation on $[N]$. The node to id mapping is known to all nodes in the system. Define the *successor* of a node v, denoted $succ(v)$, as the node u whose identity is $(id(u) + 1) \mod N$.

Virtual routes are maintained from every node to its k successors in the identity space, where k is a parameter of the scheme. In our analysis to simplify things we assume that $k = 1$, i.e. the virtual topology is just the ring. For identities i, j, define $dist(i, j)$ to be the number of edges in the shortest path from i to j in this virtual overlay network. Thus for the ring case $k = 1$, $dist(i, j)$ is $j - i$ if $i < j$, and $N - (i - j)$ otherwise. In the simulations we tested the case of larger k.

The virtual topology induces a virtual path between every two nodes. These paths are realized in the physical network via a set of predetermined physical segments between each node u and $succ(u)$. These actual physical paths are ideally shortest paths but are not necessarily so. Denote the nodes in this physical path as $PS(u, succ(u))$. Now, every node v has a local *routing table* with entries $\langle dst, nxt \rangle$ for each path $PS(w, succ(w))$ that contains v (with $dst = succ(w)$), such that nxt is the next hop after v in the segment $PS(w, succ(w))$. The method in which these paths are chosen and maintained is not within the scope of this paper. The work in [4],[5] suggests ways of choosing these paths and argues they are easy to maintain in the face of insertions and deletions.

VRR employs a *greedy routing* (GR) strategy over the virtual identity space. When a message with destination T is injected at a source S, an initial packet header $\langle target : T, intermediate\_target : succ(S) \rangle$ is formed.

When a node u receives a packet with header $\langle T, IT \rangle$, it performs the following:

- If u has a routing-table entry $\langle T', h' \rangle$, such that T' is closer to T than IT (in the virtual distance $dist(\cdot, T)$), then u modifies the header by overwriting intermediate-target with T'. If there is more than one such T', u picks the one closest to T. It forwards the packet to h'.
- Otherwise, u forwards to h, where $\langle IT, h \rangle$ appears in u's routing table.

The entire node-to-node routing path is called the *actual routing path*. In this work, we are interested in analyzing the expected length of the actual routing path, over the choices of identities for a variety of initial graphs.

3 Stretch Analysis

We first give some intuition on the routes generated by GR. Suppose that GR is invoked from s to t. The first routing table lookup performed by GR at s finds an intermediate target m_0 with identity between s and t. This intermediate

target may be $(s+1)$ or some node u such that s lies on a path $PS(u, succ(u))$ and $dist(u,t) < dist(s+1,t)$. In this case, s chooses the routing table entry corresponding to target $succ(u)$.

In this case GR continues to the next table lookup, which is invoked at a node w following s en route to m_0. Note that w must have m_0 in its routing table, or w is m_0 itself. Therefore, GR continues with an intermediate target no farther than m_0. However, a change of intermediate target may occur. First, if w is m_0, then it will find among $m_0+1, m_0+2, ..., m_0+k$ an intermediate target m_1 closer to t. We call this a *non-greedy transition*. Second, w may find in its routing table an entry m_1 closer than m_0 to t. Again, this happens when w is on a path leading to such m_1. In this case, GR moves to a route leading towards m_1. We call this a *greedy transition*.

The route to m_1 may get interrupted again, and so on. Finally, a route to the target t itself will be found, at which point the intermediate target becomes fixed.

More formally, we have the following definition. For source-destination pair (s,t), let $D(s,t)$ denote $m_0, m_1, ..., m_c = t$ the sequence of intermediate targets set by GR. We say that a transition from m_i to m_{i+1} is *non-greedy* if it was set at m_i from amongst $m_i+1, \ldots, m_i+k$, and it is *greedy* otherwise.

We will upper bound the actual routing path length by bounding the size of $D(s,t)$ in a conservative way: if $Diam$ is the diameter of the network, the length of the actual routing path between s and t is at most $Diam \cdot |D(s,t)|$.

Generally, a greedy hop at step j may depend on the first j hops. Obviously, it must be caused by a route toward a target closer to the destination than the intermediate target at step $j-1$ is. Additionally, it must be caused by a route that does not go through any of the first $j-1$ hops. In order to handle these weak dependencies, we introduce a slight generalization of the GR procedure called GR$'$. The idea in GR$'$ is to choose a greedy hop only if this change is a significant improvement. We do this by introducing a parameter α to the first routing rule as follows:

- If u has a routing-table entry $\langle T', h' \rangle$, such that T' is closer to T than IT **by factor of α or more**, i.e. $dist(T',T)/dist(IT,T) < \alpha^{-1}$, then u modifies the header by overwriting intermediate-target with T'. If there is more than one such T', u picks the one closest to T. It forwards the packet to h'.

Note that GR corresponds to special case of GR$'$ with $\alpha = 1$.We now prove some bounds on the expected actual path length of GR$'$. First we observe that the number of greedy virtual hops is at most logarithmic.

Lemma 1. *For any source s and target t, the number of* greedy transitions *in $GR'(s,t)$ is $O(\log_\alpha(N))$.*

Proof. We bound the number of greedy transitions by observing that in a greedy transition from m_i to m_{i+1}, the destination m_{i+1} is closer to t by a factor α. Hence, after at most $\log_\alpha(dist(s,t))$ greedy transitions we reach the target.

It remains to bound the number of non-greedy transitions. Recall that $D(s,t)$ denotes the total number of intermediate targets seen by the algorithm and that we bound the path length by bounding the size of $D(s,t)$. The key observation in this section is that the bound is parameterized by the likelihood of path intersection, formally defined below.

First recall the definition of intersection coefficient. We refer to a physical segment between two points chosen uniformly at random as a random virtual hop.

Definition 1. *Let $p = p(N)$ be such that two independent random virtual hops intersect with probability p. We say that the* intersection coefficient *of the set of paths is p.*

Now suppose that we were concerned about the probability that a random virtual hop intersects at least one of l other independent random virtual hops. The following definition defines conditions under which such probabilities can be estimated.

Definition 2. *A set of virtual hops $(s_1,t_1),\ldots,(s_l,t_l)$ are said to be* almost mutually exclusive *if for a random virtual hop (s,t), the probability that $PS(s,t)$ intersects one of the paths $PS(s_i,t_i)$ is at least $\frac{1}{2}lp$.*

Note that the expected number of i's such that $PS(s,t)$ intersects $PS(s_i,t_i)$ is in fact lp. However, these events are not independent, and not mutually exclusive.

Definition 3. *Let $p = p(N)$ be such that for a constant c and for any $l \in [1, \frac{1}{cp}]$, the probability that ℓ random virtual hops are not almost mutually exclusive is at most polynomially small in the size of the network. Then we say that the* group intersection coefficient *is p.*

Lemma 2. *With high probability, for all pairs s,t it holds that $|D(s,t)|$ can be bounded by $O(\frac{\alpha\log(1/p)+\log_\alpha N}{\sqrt{p}})$, where the probability is taken over the choice of mapping id's to nodes.*

Proof. We first give some intuition for the proof. The number of greedy hops is clearly at most $O(\log_\alpha N)$. Consider the $\frac{1}{\sqrt{p}}$ virtual hops $(t-j-1,t-j)$ for $j \in [1,\frac{1}{\sqrt{p}}]$ closest to the destination t in the ring. If we reached one of these virtual hops within the first $\frac{1}{\sqrt{p}}$ non-greedy hops in the routing, then we would get a bound of $O(\frac{1}{\sqrt{p}}+\log_\alpha N)$ on $|D(s,t)|$. What is the likelihood that the first $\frac{1}{\sqrt{p}}$ non-greedy hops in $D(s,t)$ do not reach this set? For this to happen, each of the $\frac{1}{\sqrt{p}}$ completed non-greedy hops must avoid hitting one of the $\frac{1}{\sqrt{p}}$ virtual hops close to the destination. Since this gives us $O(\frac{1}{p})$ pairs of virtual hops, and each pair intersects with probability p, this avoidance is unlikely. Of course there are dependencies to be taken care of and we formalize the argument below.

For any r, call the virtual hops $(t-j-1,t-j)$ for $j \in [1,r]$ the *r-last hops*. Let k,l be parameters to be chosen later. We will argue that with high probability,

within l virtual hops when routing from s to t, the current intermediate target is within distance αk in the virtual space.

Let $m_0, m_1, \ldots, m_l$ be the sequence of first l routing destinations set by GR. Of these, some $r < \log_\alpha N$ are chosen due to a greedy hop, let these be $m_{i_1}, \ldots, m_{i_r}$. Let a configuration C be defined by a set of at most $\log_\alpha N$ indices $i_1, \ldots, i_r$. For a fixed configuration C, we shall bound the probability that any sequence $m_0, m_1, \ldots, m_l$ has not hit the αk-last hops.

Let $D'(s,t)$ be the list of m_i's such that $i \notin \{i_j, i_j - 1, i_j + 1\}$. Let $l' = \lfloor \frac{1}{2}|D'(s,t)| \rfloor$. Clearly, $l' > \frac{l}{3} - 3\log_\alpha N$. $D'(s,t)$ contains at least l' disjoint pairs $(m_i, m_i + 1)$ such that $m_i - 1, m_i, m_i + 1, m_i + 2$ are all in $D(s,t)$.

The k-last hops consists of $k/2$ disjoint virtual hops. Thus except with polynomially small probability, these $k/2$ virtual hops are almost-mutually exclusive. Thus the virtual hop (m_i, m_{i+1}) intersects one of the the k-last hops with probability at least $kp/4$ (k is taken to be smaller than $\frac{1}{cp}$). Moreover, this event for (m_i, m_{i+1}) depends only on the random assignment of the virtual identifies m_i and m_{i+1} to physical nodes, and is therefore independent of the corresponding event for $(m_j, m_j + 1)$, for any $j : |j - i| > 1$.[3] Thus the probability that for a fixed configuration C, a prefix $m_0, \ldots, m_l$ exists that satisfies C but does not intersect the k-last hops is bounded by

$$(1 - \frac{kp}{4})^{l'}$$

Unless the prefix $m_0, \ldots, m_l$ has already hit the αk-last hops, any intersection with the k-last hops is a greedy step that GR' would have taken. Thus the above bounds the probability that for a fixed C, the prefix $m_0, \ldots, m_l$ defined by C has not reached the αk-last hops.

We next bound the number of configurations. There are $\binom{l}{r}$ ways of choosing the indices $i_1, \ldots, i_r$, and since $r \leq \log_\alpha N$, the number of configurations is at most

$$\log_\alpha N \binom{l}{\log_\alpha N}.$$

On the other hand, for $|D(s,t)|$ to be greater than $l + \alpha k$, the prefix $m_0, \ldots, m_l$ must not hit the αk-last hops. Thus the probability

$$Pr[|D(s,t)| > l + \alpha k)] \leq \log_\alpha N \binom{l}{\log_\alpha N} (1 - \frac{lp}{4})^{l/3 - 3\log_\alpha N}.$$

The claim follows by plugging in the value of $l = O(\frac{\log_\alpha N}{\sqrt{p}})$ and $k = O(\frac{\log(1/p)}{\sqrt{p}})$. □

[3] There is in fact a small dependency here: since the mapping is a permutation, m_i cannot be mapped to the same location as m_j. However, this excludes at most $O(l)$ locations for m_i and m_{i+1}, and hence conditioning changes the probabilities by at most a $(1 - \frac{l}{N})$ factor, which is negligible and ignored for the rest of the proof.

Properties of the d-dimensional Grid

In this section we identify the intersection coefficient of the grid for a natural set of paths. Consider a d-dimensional grid with n^d nodes, each node can be identified by a d-dimensional vector in $[0, n-1]^d$. Let $\boldsymbol{s} = (s_1, \ldots, s_d)$ and $\boldsymbol{t} = (t_1, \ldots, t_d)$ be two nodes. There are many shortest paths between them. Natural candidates for a collection of paths are paths that follow more or less the l_2 shortest path between the points. The paths we analyze, and use to drive the simulation are crude approximations. We randomly sample an intermediate node $\boldsymbol{w} = (w_1, \ldots, w_d)$ where each w_i is uniformly sampled in $[s_i, t_i]$ and then route through $\boldsymbol{w}$ as follows: first route from from $\boldsymbol{s}$ to $\boldsymbol{w}$ by fixing the coordinates one after the other, i.e. first go to $(w_1, s_2, \ldots, s_d)$ and so on. Once $\boldsymbol{w}$ is reached, route to $\boldsymbol{t}$ by fixing the coordinates in reverse order, i.e. first go to $(w_1, w_2, \ldots, t_d)$ and so on. The node $\boldsymbol{w}$ is called the *intermediate routing node* of the path. Denote by $p(c) = cn^{-(d-2)}$. The next bound states that there is a way to chose c as a function of d such that p is the intersection coefficient of the network.

Lemma 3. *For every d there is c such that for every n, the intersection coefficient of the n^d grid is $p = cn^{-(d-2)}$.*

Proof. The proof is by induction on d. For $d = 2$, $p(c) = c$ so we need to show that the probability two virtual hops intersect is at least a constant independent of n. Intuitively this should hold because with constant probability both paths are roughly diagonals in the two-dimensional grid and thus intersect. A formal (and rather crude) argument is as follows: say the first source–target pair is $(s_1^{(1)}, s_2^{(1)})$ and $(t_1^{(1)}, t_2^{(1)})$. Similarly the second pair is $(s_1^{(2)}, s_2^{(2)})$ and $(t_1^{(2)}, t_2^{(2)})$. With probability 3^{-6} it holds that $s_1^{(1)}, s_2^{(1)} \leq n/3$ and $t_1^{(1)}, t_2^{(1)} \geq 2n/3$, and their intermediate node $(w_1^{(1)}, w_2^{(1)})$ satisfies that $w_1^{(1)}, w_2^{(1)} \in [n/3, 2n/3]$. In other words the path $\boldsymbol{s}^{(1)} \to \boldsymbol{t}^{(1)}$ is a diagonal. Similarly with probability 3^{-6} the path $\boldsymbol{s}^{(2)} \to \boldsymbol{t}^{(2)}$ is the crossing diagonal, i.e. $s_1^{(2)}, t_2^{(2)} \leq n/3$ and $t_1^{(2)}, s_2^{(2)} \geq 2n/3$, and $w_1^{(2)}, w_2^{(2)} \in [n/3, 2n/3]$. If both these events occur then the paths intersect.

Now assume $d > 2$. Let $w^{(1)}$ and $w^{(2)}$ denote the intermediate hops. If $w^{(1)}$ and $w^{(2)}$ agree in the first $(d-2)$ co-ordinates, then the probability of intersection is at least c using the two-dimensional case. Since $w^{(1)}$ and $w^{(2)}$ are drawn from the same probability distribution, the probability that they agree on the first $(d-2)$ co-ordinates is at least $n^{-(d-2)}$: the collision probability for a distribution is maximized when it is uniform, in which case we get the $n^{-(d-2)}$ bound. Moreover, it is easy to check that the collision probability is at least $(an)^{-(d-2)}$ for a constant a. The claim follows. □

Lemma 4. *For every d there is c such that for every n, the group intersection coefficient of the n^d grid is $p = cn^{-(d-2)}$.*

Proof. The proof is very similar to the previous lemma. For $d = 2$, there is nothing to prove since l is at most 1.

Now assume $d > 2$. Let $w^{(1)}, \ldots, w^{(l)}$ denote the intermediate hops of the l virtual hops, and let $w^{(*)}$ denote the intermediate hop for (s, t). If $w^{(*)}$ agrees

with one of the $w^{(i)}$'s in the first $(d-2)$ co-ordinates, then the probability of intersection is at least c using the two-dimensional case. Since $w^{(1)}, \ldots, w^{(l)}$ are drawn from the same probability distribution and $l < \frac{1}{cp}$ they span at least $l/2$ different values for the first $(d-2)$ co-ordinates with high probability. The claim follows. □

Lower Bound

Lemma 5. *If* $dist(s,t) = \frac{1}{10\sqrt{p}}$ *then with probability at least* 0.99 *the size of* $D(s,t)$ *is at least* $\frac{1}{10\sqrt{p}}$.

Proof. We calculate the probability there is a greedy hop in the path. In total there are $\frac{1}{100p}$ pairs of paths. Each of them intersects with probability p so on expectation there are $1/100$ intersections. Markov's inequality implies that the probability there is at least one greedy hop is at most 0.01. □

We can also show the following result, the proof of which is omitted from this extended abstract:

Lemma 6. *For* $1 < s < t < N-1$, *we have*

$$E[D(s,t+1)] \geq \min\{E[D(s,t)], \frac{1}{10\sqrt{p}}\}.$$

The above two lemmas imply that for randomly chosen s and t the expected size of $D(s,t)$ is $\Omega(\frac{1}{\sqrt{p}})$. Thus our upper bound is tight up to logarithmic factors.

4 Simulation

We simulate a family of VRR schemes over d-dimensional grids. The challenging aspect of our simulation is scaling. In order to demonstrate asymptotic trends, we want to test networks of considerable sizes. This cannot be done naively. The fundamental routing step in VRR scheme involves a routing-table lookup. Naively simulated, this requires maintaining $O(n\times$ routing table size) information. For some of the topologies we consider, this prohibits simulating networks beyond 2^{20} nodes ($\approx 2^{30}$ entries $\approx$ 8GB memory). Though this is already quite sizable, we devised a simulation technique that can scale even higher. We first describe the simulation technique, and then present the result.

4.1 Simulation Framework

The underlying (physical) networks in our simulation are d-dimensional grids with n nodes on each side (and $N = n^d$ nodes in total). Nodes have integral physical identifiers from 0 to $n^d - 1$, assigned so as to allow a node's position in the grid to be easily retrieved from its identifier (and vice-versa).

In our simulation, we take $sp(u, v)$, a shortest path, as the physical segment $PS(u, v)$ between neighboring nodes u and v in the virtual space. In general, these paths are not unique in a d-dimensional cube; we pick paths that we analyzed in the previous section.

Suppose we are computing the route to a (virtual) target t and let v be the current vertex. VRR schemes need to examine v's routing table and find the intermediate target t' that is the closest (in the virtual space) predecessor of t in the ring.

Memory constraints prevent us from storing the routing tables explicitly when simulating very large networks. Instead, we check all possible candidates for t' (starting at t, then $t-1$, then $t-2$, and so on) until we find one that would actually be an intermediate target in v's routing table. For each candidate t', we must check if there is a physical segment that crosses v. Such a segment would have as endpoints t' and a virtual neighbor of t', denoted by s'. Let $S(t')$ be the set of virtual sources s' such that (s', t') is a physical segment. Note that $S(t')$ is just $\{t'-1\}$ in a simple ring, but in other overlay topologies we experiment with, it is a set. For each $s' \in S(t')$, we can check in $O(d)$ time whether v belongs to the physical segment from s' to t'. If it does, we can stop: t' is the best entry in v's routing table.

Implicit mapping. Even storing the mapping of nodes to virtual identities (with quick reverse lookup) is quite costly for sizable networks, and we avoid that. Our simulation picks as the virtual identifiers a (pseudo-)random permutation of $[0, n^d - 1]$. We do not maintain the permutation explicitly in memory. Instead, we keep it implicitly with the Luby-Rackoff scheme [9], which works as follows.

Assume node identifiers have exactly $2k$ bits (i.e., $N = 2^{2k}$). We must define a permutation $\pi : \{0,1\}^{2k} \to \{0,1\}^{2k}$, so that a node with physical identifier x has virtual identifier $\pi(x)$. An identifier $x = (L, R)$ can be seen as the concatenation of its first k bits (L) and last k bits (R). Define $\pi(x)$ as $\pi(L, R) = (R, f(R) \oplus L)$, where $f : \{0,1\}^k \to \{0,1\}^k$ is an auxiliary pseudorandom function. It is easy to see that $\pi(x)$ produces a permutation of all $2k$-bit strings. When f is sampled from a family of one-way functions, Luby and Rackoff proved that it suffices to iterate π four times, sampling a fresh function each time, to obtain a pseudo-random permutation. Therefore, to convert a physical identifier x into the corresponding virtual identifier, we simply compute $\pi^*(x) = \pi(\pi(\pi(\pi(x))))$. To convert a virtual identifier to a physical identifier, we use the inverse function $\pi^{-1}(x) = \pi^{-1}(L, R) = (f(L) \oplus R, L)$, also iterated four times.

To determine $f(X)$ (where X is a k-bit string), our implementation concatenates X with a user-defined 32-bit seed s, calculates the 128-bit MD5 hash of the resulting string, and discards all but the first k bits of the result. These operations (in particular the MD5 computation) are costly in practice. To speed up the simulation, we use two levels of caching. We remember the first C pairs $(x, \pi^*(x))$ that we evaluate, as well as the result of every $f(X)$ computation we ever perform (the corresponding table, with $\sqrt{N}$ entries, is small enough to fit in memory). We used $C = 2.5 \cdot 10^8$ in our experiments.

4.2 Results

We start our experiments with the most basic version of VRR, in which each vertex has a single virtual neighbor in the ring (i.e., $|S(v)| = 1$ for every v). Table 1 shows the results obtained for grids with 2, 3, and 4 dimensions and various sizes. Every entry in the table was computed from 1000 routes. The endpoints of each route are two nodes picked uniformly at random. Each route uses a different pseudorandom mapping between physical and virtual nodes.

Table 1. Simple ring simulation results. Columns are: Grid dimensionality; network size; average nodes on shortest path; average nodes on actual routing path; 99th percentile of nodes on actual routing path; 99th percentile stretch; aggregate stretch.

DIM	NODES	SHRT.	NODES		STRETCH	
		PATH	AVG	99TH	99TH	AGGR.
2	2^{24}	2685	7679	18548	18.75	2.86
	2^{26}	5539	15494	37511	26.37	2.80
	2^{28}	10720	31141	70889	23.36	2.90
	2^{30}	22113	61979	141664	21.80	2.80
	2^{32}	43456	120237	300127	23.02	2.77
3	2^{12}	16	94	212	34.01	5.83
	2^{18}	65	736	1714	51.00	11.35
	2^{24}	255	5797	13986	112.76	22.75
	2^{30}	1031	45180	111140	270.81	43.82
4	2^{20}	42	1316	3200	118.05	31.23
	2^{24}	86	5295	12844	301.34	61.86
	2^{28}	170	21087	49399	504.08	123.89
	2^{32}	342	85614	193431	1002.67	250.10

For each instance size, we report the average shortest path length and the average actual routing path length. The ratio between these two is the *aggregate stretch*, which is our main performance measure and is reported in the last column. For reference, we also report the 99th percentile of the actual path length (over all 1000 routes).

Recall that the intersection coefficient is $p \in O(N^{-\frac{d-2}{d}})$, and the expected stretch is proportional to $1/\sqrt{p}$. Hence, for the two-dimensional case, the expected stretch is constant; for $d = 3$, when we grow n by a factor of 2^6, we expect the average stretch to grow by a factor $(2^6)^{1/6} = 2$; and for $d = 4$, when growing n by factor 2^4, the stretch is expected to grow by factor $(2^4)^{1/4} = 2$. Table 1 indeed demonstrates these trends.

Other Overlay Structures. We considered two variations of the simple VRR ring. As suggested in the original VRR work, we vary the number c of ring successors to which each node maintains connections.

Additionally, we considered a variation in which the overlay topology has sublinear hop diameter. The idea here is that even without the effect of path intersection and greedy hops, the stretch is bounded by the routing complexity of the overlay network. Specifically, we introduce a small change, one that would not impair the spirit and practical value of VRR scheme. Borrowing from *small world* extensions of ring overlays [3], we replace the c-th neighbor of a node v with $v - 2^j$, where j is an integer picked uniformly at random from the range $[\lceil \log_2 c \rceil, (\log_2 N) - 1]$.

Table 2 shows the average aggregate stretch (over 1000 seeds) for the various topologies.

Table 2. Simulation results with varying overlay topologies, with 1, 2, and 5 virtual neighbors. The small world variations are denoted with a '$*$'.

NETWORK		NEIGHBORHOOD SIZE				
DIM	NODES	1	2	2^*	5	5^*
2	2^{24}	2.86	2.24	2.25	1.73	1.68
	2^{26}	2.80	2.16	2.16	1.76	1.66
	2^{28}	2.90	2.24	2.29	1.82	1.74
	2^{30}	2.80	2.23	2.20	1.78	1.72
	2^{32}	2.77	2.20	2.23	1.80	1.74
3	2^{12}	5.83	3.72	3.99	2.47	2.42
	2^{18}	11.35	6.90	7.45	4.01	3.99
	2^{24}	22.75	13.34	13.72	7.30	7.21
	2^{30}	43.82	26.42	24.10	12.51	12.69
4	2^{20}	31.23	18.55	17.72	8.98	8.81
	2^{24}	61.86	34.56	27.70	16.46	15.46
	2^{28}	123.89	69.58	42.73	31.93	27.27
	2^{32}	250.10	136.60	66.98	63.12	48.26

Increasing the size of virtual neighborhoods reduces the average stretch, since the intersection coefficient increases correspondingly. The asymptotic trends remain the same with increased neighborhood sizes (on a simple ring). The effect of small world links become noticeable only with four dimensions, and fairly large network size. This is when the polylogarithmic effect of small world routing starts dominating the simple ring stretch of $O(N^{d-2/2d}) = O(N^{1/4})$.

Modified Greedy Routing. Finally, we examine the effect of modifying the greedy hop criterion as suggested in our analysis section above. We introduce into the VRR scheme a parameter α, and allow a greedy hop to occur only when the intermediate routing target is improved by a factor α. When $\alpha = 1$ (as in the experiments reported so far), the routing algorithm performs every greedy step it can. For larger values of α, a greedy step (i.e., a change of the intermediate

target) happens only if the gap to the final target (in the virtual space) is reduced by factor α.

Table 3 shows the average aggregate stretch (over 1000 routes) for three values of α: 20, 2, and 1. The results show that setting α to 2 has little effect on the performance of the routing algorithm. begintable[thbp]

Table 3. Average aggregate stretch with different α values.

NETWORK		GREEDY FACTOR (α)		
DIM	NODES	20	2	1
2	2^{24}	3.67	2.95	2.86
	2^{26}	3.62	2.86	2.80
	2^{28}	3.65	2.96	2.90
	2^{30}	3.69	2.92	2.80
	2^{32}	3.54	2.80	2.77
3	2^{12}	8.31	6.10	5.83
	2^{18}	18.14	12.63	11.35
	2^{24}	38.63	25.14	22.75
	2^{30}	77.17	49.97	43.82
4	2^{20}	55.14	34.11	31.23
	2^{24}	108.56	68.85	61.86
	2^{28}	218.14	138.10	123.89
	2^{32}	426.98	279.61	250.10

As a final note, while running the experiments above, we observed that in a typical path most of the hops in a route are close (in the virtual space) to the target. Let the *median target* of a route with h nodes be the intermediate target of the algorithm when the $(h/2)$-th node is visited. With $\alpha = 1$, on average the median target was less than $\log_2 N$ hops away from the target in two dimensions. Even in higher dimensions, the average gap was always smaller than $\log^2 N$.

5 Conclusions

We have theoretically and empirically analyzed Virtual Ring Routing. We show that for a 2-dimensional grid, VRR indeed gives expected path length which is at most $O(\log N)$ times the diameter. On the other hand, for a d-dimensional grid, we show that the expected path length is at least $\Omega(N^{\frac{d-2}{2d}})$ times the diameter of the graph. We note that for the two-dimensional case, our bound only shows a bound of $O(Diam \cdot \log N)$ on the routing path length. Empirically, VRR does not seem to exhibit good *locality* properties. It would be interesting to investigate this question further.

References

1. I. Abraham, C. Gavoille, and D. Malkhi. On space-stretch trade-offs: Upper bounds. In *ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, July 2006.
2. I. Abraham, C. Gavoille, and D. Malkhi. Compact routing for graphs excluding a fixed minor. In *19th Intl. Symposium on Distributed Computing (DISC 05)*, September 2005.
3. L. Barrière, P. Fraigniaud, E. Kranakis, and D. Krizanc. Efficient routing in networks with long range contacts. In *DISC '01: Proceedings of the 15th International Conference on Distributed Computing*, pages 270–284, London, UK, 2001. Springer-Verlag.
4. M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron. Virtual ring routing: Network routing inspired by DHTs. In *ACM annual conference of the Special Interest Group on Data Communication (SIGCOMM)*, pages 351–362, 2006.
5. M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S. Shenker. ROFL: Routing on flat labels. In *ACM annual conference of the Special Interest Group on Data Communication (SIGCOMM)*, September 2006.
6. B.-N. Cheng, M. Yuksel, and S. Kalyanaraman. Orthogonal rendezvous routing protocol for wireless mesh networks. In *IEEE International Conference on Network Protocols*, 2006.
7. C. Gavoille. Routing in distributed networks: Overview and open problems. *ACM SIGACT News - Distributed Computing Column*, 32(1):36–52, March 2001.
8. C. Kim, M. Caesar, and J. Rexford. Floodless in SEATTLE: a scalable ethernet architecture for large enterprises.
9. M. Luby and C. Rackoff. How to construct pseudorandom permutations and pseudorandom functions.
10. Y. Mao, F. Wang, L. Qiu, S. S. Lam, and J. M. Smith. S4: Small state and small stretch routing protocol for large wireless sensor networks. In *4th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2007.
11. H. Pucha, S. M. Das, and Y. C. Hu. Imposed route reuse in ad hoc network routing protocols using structured peer-to-peer overlay routing. *IEEE Transactions on Parallel and Distributed Systems*, 2006.
12. C. Westphal and J. Kempf. A compact routing architecture for mobility. In *MobiArch '08: Proceedings of the 3rd international workshop on Mobility in the evolving internet architecture*, pages 1–6, New York, NY, USA, 2008. ACM.