A Compact Routing Scheme and Approximate Distance Oracle for Power-Law Graphs

Wei Chen[1], Christian Sommer[2], Shang-Hua Teng[3], and Yajun Wang[1]

[1] Microsoft Research Asia, Beijing, China
[2] The University of Tokyo and National Institute of Informatics, Tokyo, Japan
[3] Department of Computer Science, University of Southern California, CA, USA

Microsoft Research Technical Report

MSR-TR-2009-84

July 2009

Abstract. Compact routing addresses the tradeoff between table sizes and stretch, which is the worst-case ratio between the length of the path a packet is routed through by the scheme and the length of a shortest path from source to destination. We adapt the compact routing scheme by Thorup and Zwick to optimize it for power-law graphs. We analyze our adapted routing scheme based on the theory of unweighted random power-law graphs with fixed expected degree sequence by Aiello, Chung, and Lu. Our result is the first theoretical bound coupled to the parameter of the power-law graph model for a compact routing scheme. In particular, we prove that, for stretch 3, instead of routing tables with $\tilde{O}(n^{1/2})$ bits as in the general scheme by Thorup and Zwick, expected sizes of $O(n^{\gamma} \log n)$ bits are sufficient, and that all the routing tables can be constructed at once in expected time $O(n^{1+\gamma} \log n)$, with $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$, where $\tau \in (2, 3)$ is the power-law exponent and $\varepsilon > 0$ (which implies $\varepsilon < \gamma < 1/3 + \varepsilon$). Both bounds also hold with probability at least $1 - 1/n$ (independent of ε). The routing scheme is a labeled scheme, requiring a stretch-5 handshaking step and using addresses and message headers with $O(\log n \log \log n)$ bits, with probability at least $1 - o(1)$. We further demonstrate the effectiveness of our scheme by simulations on real-world graphs as well as synthetic power-law graphs. With the same techniques as for the compact routing scheme, we also adapt the approximate distance oracle by Thorup and Zwick for stretch 3 and obtain a new upper bound of expected $\tilde{O}(n^{1+\gamma})$ for space and preprocessing for random power-law graphs.

1 Introduction

Message routing is a fundamental service in communication networks. When routing a message from a source to a destination in the network, to decide where to forward the message to, a node may only use its local information, which includes its local routing table, the destination address, and a message header. A routing scheme is expected to route messages between all source-destination pairs along shortest or approximate shortest paths. A key measure of the quality of a routing scheme is its worst-case multiplicative *stretch*, which is defined as the maximum ratio of the length of the message route between a pair of nodes s and t by the scheme and the actual shortest path length between s and t, among all s-t pairs in the network.

Routing schemes address the tradeoff between stretch and routing table size. A trivial stretch-1 routing scheme is one in which every node stores for every destination in the network where to forward the message to. However, for a network with n nodes, this approach requires unscalable $\Omega(n \log n)$-bit routing tables for every node [21]. A *compact* routing scheme is only allowed to have routing tables with sizes sublinear in n and message header sizes polylogarithmic in n. There are two classes of compact routing schemes: *Labeled* schemes are allowed to add labels to node addresses to encode useful information for routing purposes, where each label has length at most polylogarithmic in n. *Name-independent* schemes do not allow the renaming of node addresses, instead they must function with all possible addresses.

Both labeled and name-independent compact routing schemes have been studied extensively. Universal schemes work for all network topologies [3–5, 14, 35, 36]. It has been shown that with $\tilde{O}(n^{1/k})$-bit routing tables (as usual, we abbreviate $O(f(n) \cdot \log^t n)$ for some constant t by $\tilde{O}(f(n))$) one can achieve a stretch of $O(k)$, and that this tradeoff is essentially tight due to a girth conjecture by Erdős.

Due to these impeding lower bounds for general graphs, specialized schemes were designed for various families of network topologies, including trees [19, 24, 36], planar graphs [20, 27], fixed-minor-free graphs [2], or graphs with low doubling dimension [1, 22, 23]. These topology-specific schemes achieve significant improvements on the stretch-space tradeoff over universal routing schemes.

Power-law graphs [31] constitute an important family of networks appearing in various real-world scenarios such as the Internet, the World Wide Web, collaboration networks, and social networks [12, 18]. In a power-law graph, the number of nodes with degree x is proportional to $x^{-\tau}$, for some constant τ. The power-law exponent τ for many real-world networks is in the range between 2 and 3. Power-law graphs do not seem to belong to any of the well-studied network families such as trees, planar graphs or low doubling dimension graphs mentioned above.

Despite their high relevance in practice, the family of power-law graphs has not received much attention from the compact routing community. There are experimental studies of compact routing in power-law graphs and Internet-like graphs. Krioukov et al. [25] evaluate the universal routing scheme of Thorup and Zwick (*TZ*) [36] on random power-law graphs [6] and provide experimental evidence of much better performance (both in terms of stretch and table sizes) than the theoretical worst-case bound. However, they do not provide a theoretical bound of the TZ scheme on power-law graphs for neither stretch nor table size. Enahescu et al. [16] propose a landmark selection scheme that adapts the TZ scheme and they show empirically that their adaptation achieves good stretch and table sizes for power-law graphs and Internet Autonomous System (AS) graphs. Unfortunately, their theoretical analysis is for Erdős-Rényi random graphs [17] instead of power-law graphs. Brady and Cowen [8] give a compact routing scheme tailored for power-law graphs with additive stretch d and header and table sizes $O(e \log^2 n)$, where both d and e depend on the graph, and they show experimentally that these values are reasonably small for certain random power-law graphs [6]. However, there is no rigorous analysis connecting d and e to the parameter τ of power-law graphs.

1.1 Our Contribution

In this paper, we bridge the gap in the study of compact routing schemes for power-law graphs. We provide the first theoretical analysis that directly links the power-law exponent τ of a random power-law graph to the bound on the routing table sizes.

More specifically, we adapt the labeled universal compact routing scheme of Thorup and Zwick [36] to optimize it for unweighted, undirected power-law graphs. Our adaptations include (a) selecting nodes with the largest degrees as the landmarks instead of random sampling, and (b) directly encoding shortest paths in node labels and message headers instead of relying on a tree routing scheme (a detailed comparison with [36] is deferred to Section 1.2).

Our complexity analysis of the routing scheme is based on the random power-law graph model with expected degree sequence proposed by Aiello, Chung and Lu [6, 10, 11, 28] with some minor simplifications. We assume the power-law exponent τ to lie in the range of $(2, 3)$, which is the so called "finite mean infinite variance" region of the power-law degree distribution, where most practical power-law networks are assumed to be in.

We prove that for a stretch upper bound of 3, instead of tables of size $\tilde{O}(n^{1/2})$ shown to be optimal up to a polylogarithmic factor for general graphs [36], expected sizes of $O(n^\gamma \log n)$ bits are sufficient, and that the routing tables can be constructed at once in expected time $O(n^{1+\gamma} \log n)$, with $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$ and $\varepsilon > 0$ (which implies $\varepsilon < \gamma < 1/3 + \varepsilon$). Both bounds also hold with probability at least $1 - 1/n$ (independent of ε). This means that for all $\tau \in (2, 3)$, we have an upper bound of $\tilde{O}(n^{1/3+\varepsilon})$ on the routing table sizes, which is better than the optimal bound of $\tilde{O}(n^{1/2})$ for general graphs. For values of τ close to 2, for example for $\tau = 2.1$, which is the exponent that fits the power-law distribution well to the degree distribution of the actual Internet inter-domain graph [18, 25], our bound is $O(n^{1/12+\varepsilon})$, which indicates that the adapted TZ routing scheme would be very effective on Internet-like graphs. The routing scheme requires a stretch-5 handshaking (similar to [36, Sec. 4]), and uses addresses and message headers of size $O(\log n \log\log n)$, with probability at least $1 - o(1)$. The efficient encoding using $O(\log n \log\log n)$ bits in addresses and headers relies on specific distance properties of power-law graphs. Our scheme is a *fixed-port* scheme, meaning that it works for any permutation of port number assignments on any node.

We provide simulation results for both random power-law graphs and actual router-level networks, which demonstrate the effectiveness of our adapted compact routing scheme (Section 5).

Using the same techniques, we also adapt the approximate distance oracle by Thorup and Zwick [37] for unweighted, undirected power-law graphs. We prove that, for stretch 3, instead of an oracle of size $O(n^{3/2})$, expected space $O(n^{1+\gamma})$ is sufficient and that the oracle can be constructed in expected time $O(n^{1+\gamma} \log n)$. Again, both bounds also hold with probability at least $1 - 1/n$.

1.2 Additional Details on Related Work

We provide additional details on the comparison with Thorup and Zwick's routing schemes, and on other random power-law graph models.

Thorup and Zwick [36] contribute two different routing schemes. Their first scheme is a stretch-3 scheme with an $O(n^{1/2} \log^{3/2} n)$-bit routing table per node and $O(\log n)$-bit labels and headers. This scheme is based on Cowen's earlier scheme [14], which uses a small subset A of nodes, called *landmarks*, to route messages. In a graph $G = (V, E)$, for every node u, define its *cluster* $C(u) = \{v \in V \ : \ d(v, u) < d(v, A)\}$, where $d(v, u)$ and $d(v, A)$ denote the graph distance from v to u and A, respectively. Let $\ell(u)$ denote the landmark in A that is the closest to node u (ties are resolved arbitrarily). The routing table of node u stores the port identifiers to route messages to all nodes in A and $C(u)$. If a destination v is not in $A \cup C(u)$, u routes through $\ell(v)$, which guarantees a stretch bound of 3 due to the definition of the cluster $C(u)$. Thorup and Zwick use a resampling method to achieve $|A \cup C(u)| = O(n^{1/2} \log^{1/2} n)$ for every node u.

The second scheme of Thorup and Zwick [36] is based on their approximate distance oracle [37]. For any $k \geq 2$, they design a compact routing scheme with $\tilde{O}(n^{1/k})$-bit tables, $O(k \log^2 n / \log\log n)$-bit addresses, and $O(\log^2 n / \log\log n)$-bit headers (the bounds on addresses and headers are for fixed-port schemes). The scheme achieves stretch $2k-1$ with a stretch $4k-5$ handshake. For the case of $k = 2$ (comparable to our scheme), their scheme essentially considers the landmark set A together with the *ball* of a node u, $B(u) = \{v \ : \ d(v, u) < d(u, A)\}$. Note that balls and clusters are dual concepts: $v \in C(u)$ if and only if $u \in B(v)$. The routing table of u stores the ports to route messages to all nodes in $A \cup B(u)$. Similar to the first scheme, when $v \notin A \cup B(u)$, u routes through $\ell(v)$ to reach v, but in this case it only guarantees a stretch of 5 instead of 3 when $v \notin B(u)$ but $u \in B(v)$. A *handshake* is needed to reduce the stretch to 3. Moreover, a node w on the path from $\ell(v)$ to v may not know the port to route to v from its routing table, since v may not be in $B(w)$ (though $v \in C(w)$). To resolve this issue, Thorup and Zwick further use a tree routing scheme, which requires additional, rather complicated labels. They use random sampling to guarantee that $|A \cup B(u)| = \tilde{O}(n^{1/2})$.

Our scheme is similar to their second scheme. We also use balls and landmarks to route messages. There are two major differences: First, we use high-degree nodes instead of randomly selected nodes as landmarks. The major contribution of the paper is to prove that, with this selection strategy, in random power-law graphs, we achieve $|A \cup B(u)| = O(n^\gamma)$ with $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$ and $\varepsilon > 0$, which holds both in expectation and with high probability. Second, instead of using a tree routing scheme, we directly encode the shortest path from $\ell(v)$ to v in v's address, which is short (with probability $1 - o(1)$) due to the distance properties in random power-law graphs. As a result, our routing table sizes are smaller than the tables in both TZ schemes, and our address and header size of $O(\log n \log\log n)$ is better than the second scheme and close to the first scheme. Our scheme is also simpler than the second scheme and is comparable with the first scheme. This improvement is possible only by tailoring the scheme to unweighted power-law graphs.

Besides the random power-law graph model of Aiello, Chung, and Lu [6, 10, 11, 28], other mathematical models for power-law graphs include the configuration model [33], the Poissonian process [34], and the preferential attachment model [7, 26]. Among these, the random power-law graph model is studied very well, providing a rich body of mathematical results. Furthermore, recent empirical studies on compact routing also use this model [8, 25].

2 Preliminaries

We adapt the random graph model for fixed expected degree sequence as defined by Aiello, Chung, and Lu [6, 10, 11, 28] using the definition from [10, Section 2]. We refer to the original random graph distribution using the expression Fixed Degree Random Graph (**FDRG**).

Definition 1. *For a constant $\tau \in (2, 3)$, the random power-law graph distribution* $\mathbf{RPLG}(n, \tau)$ *is defined as follows. Let the sequence of generating parameters $\boldsymbol{w} = \{w_1, w_2, \ldots, w_n\}$ obey a power law, that is $w_k = \left(\frac{n}{k}\right)^{1/(\tau-1)}$ for $k \in \{1, 2, \ldots n\}$. The edge between v_i and v_j is inserted into the random graph with probability* $\min\{w_i w_j \rho, 1\}$*, where $\rho = \frac{1}{\sum_k w_k}$.*

Note that we adapt the original model by deterministically inserting edges if $w_i w_j > \sum_k w_k$, since in the **FDRG** model it is required that $\forall i, j : w_i w_j < \sum_k w_k$, which, without modification, rules out the values for τ we consider in this paper. In the **FDRG** model, the value w_i corresponds to the expected degree of vertex v_i, and they refer to $\boldsymbol{w}$ as the *expected degree sequence.* In our adaptation, the graph is sampled due to the generating parameter values w_i. Let D_i be the random variable denoting the degree of node v_i. In our model, the expected degree $E[D_i]$ of node v_i is smaller than or equal to the generating parameter w_i.

We require that $n = |V(G)|$ is sufficiently large, specifically, that

$$n^{\frac{\varepsilon(2\tau-3)}{\tau-1}} \geq \frac{2(\tau-1)}{\tau-2} \ln n. \tag{1}$$

Our results do not have any other implicit dependencies on ε.

The *core* of a graph consists of nodes having large degrees. Let $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$ for some $\varepsilon > 0$ and $\gamma' = \frac{1-\gamma}{\tau-1}$.

Definition 2. *For a power-law degree sequence $\boldsymbol{w}$ and a graph G with n nodes, the* core *with degree threshold $n^{\gamma'}$, $\gamma' \in (0, 1)$, is defined as follows.*

$$\mathsf{core}_{\gamma'}(\boldsymbol{w}) := \{v_i : w_i > n^{\gamma'}\},$$
$$\mathsf{core}_{\gamma'}(G) := \{v_i : \deg_G(v_i) > n^{\gamma'}/4\},$$

where $\deg_G(v_i)$ *is the degree of* v_i *in* G *(the subscript* G *is omitted when the graph is clear from the context).*

Our $\mathsf{core}_{\gamma'}(\boldsymbol{w})$ is the $n^{\gamma'}$-Core in [28, Chapter 4, Definition 2].

For each vertex u of a graph G, we define its ball relative to the core as

$$B_G(u) := \{v \in V(G) \,:\, d(u,v) < \min_{v' \in \mathsf{core}_{\gamma'}(G)} d(u,v')\}.$$

3 The Adapted Compact Routing Scheme

Let the unweighted graph $G = (V, E)$ model the network. Each node v in the network has a unique $\lceil \log_2 n \rceil$-bit static name. Whenever we write v in a routing table, a message header, or a node address, we mean its $\lceil \log_2 n \rceil$-bit static name representation. Each node v has $\deg(v)$ ports connecting it with its neighbors. These ports are numbered by $0, 1, \ldots, \deg(v) - 1$, and thus each port number of v requires $\lceil \log_2 \deg(v) \rceil$ bits. For every packet, the routing scheme needs to decide which port the packet is to be forwarded to. Our scheme is a fixed-port scheme, that is, it works with arbitrary permutations of port number assignments.

3.1 Routing Scheme

The routing algorithm is inspired by and based on [14, 36]. We also use a set of landmarks $A \subseteq V$, but different from [14, 36], we use $\mathsf{core}_{\gamma'}(G)$ as landmarks instead of nodes sampled at random. For each node u in G, let $\ell(u)$ denote u's closest landmark, that is, $\ell(u) := \arg \min_{v \in \mathsf{core}_{\gamma'}(G)} d(u,v)$. The local targets of node u are defined as the elements of its *ball* $B_G(u)$. Similar to the second scheme in [36], each node u stores the ports to route messages along the shortest paths to all landmarks and to its local targets. If the target v is neither a landmark nor a local target of u, the message is routed to v's closest landmark $\ell(v)$ and from there to the target v.

The scheme is a labeled scheme. For a node u to know $\ell(v)$ of any target v, the address of node v contains an encoding of $\ell(v)$. Moreover, for a node w on the shortest path from $\ell(v)$ to v ($w \neq \ell(v)$ and $w \neq v$), v may not be in $B_G(w)$ and thus w may not know the port to route messages to v. To resolve this issue, we further extend the address of v by *encoding* the shortest path from the landmark $\ell(v)$ to v.

Let $(s = u_0, u_1, \ldots, u_m = t)$ denote the sequence of nodes on a shortest path from s to t. Let $SP(s,t)$ be the encoding of this shortest path as an array with m entries, wherein $SP(s,t)[i]$ denotes the port to route from u_i to u_{i+1} for all $i = 0, 1, \ldots, m-1$. Thus $SP(s,t)$ can be encoded with $\sum_{i=0}^{m-1} \log_2 \lceil \deg(u_i) \rceil$ bits. We now provide the precise definitions of addresses, message headers, and local routing tables.

Definition 3.

- *The address of node* $u \in V$ *is* $\mathtt{addr}(u) := (u, \ell(u), SP(\ell(u), u))$.
- *The header of a message from node* s *to node* t *is in one of the following formats:*
  1. $\mathtt{header} = (route, s, t)$*, where* $route = \mathtt{local}$*,*
  2. $\mathtt{header} = (route, s, addr)$*, where* $route = \mathtt{toLandmark}$ *and* $addr = \mathtt{addr}(t)$*,*
  3. $\mathtt{header} = (route, s, t, pos, SP)$*, where* $route \in \{\mathtt{fromLandmark}, \mathtt{direct}\}$*,* pos *is a non-negative integer that may be modified along the route, and* $SP = SP(s,t)$ *if* $route = \mathtt{direct}$ *or* $SP = SP(\ell(t), t)$ *if* $route = \mathtt{fromLandmark}$*,*
  4. $\mathtt{header} = (route, s, t, SP)$*, where* $route = \mathtt{handshake}$ *and* SP *is a reversed shortest path from* t *to* s *to be encoded along the path from* s *to* t*.*

– *The local routing table for each node u consists of the information about routes to the core and the information about local routes:*

$$\texttt{tbl}(u) := \{(v, \texttt{port}_u(v)) : v \in \mathsf{core}_{\gamma'}(G)\} \cup \{(v, \texttt{port}_u(v)) : v \in B_G(u)\},$$

where $\texttt{port}_u(v)$ is the local port of u to route messages towards node v along some shortest path from u to v.

The routing procedure is described in Algorithm 1. It includes pseudocode for the source node s to determine the method of sending a message to target t (Lines 1–10), based on whether t is local or not and whether a shortest path to t is known due to an earlier handshake or not. It also includes pseudocode for an intermediate node u to determine whether to forward the message using its local routing table (Lines 20 and 26), or to forward the message using the shortest path encoded in the header (Lines 22–24), or to switch the routing direction from towards the landmark $\ell(t)$ to towards the target t (Lines 16–18). The correctness of the algorithm is based on the simple observation that if $t \in B_G(s) \cup \mathsf{core}_{\gamma'}(G)$ (and thus t is in the routing table of s), then, for all nodes w on the shortest path from s to t, we also have $t \in B_G(w) \cup \mathsf{core}_{\gamma'}(G)$.

Algorithm 1 LandmarkBallRouting on node u, with source s, target $t \neq s$, and header `header`.

1: **if** $u = s$ **then**
2:   **if** $t \in B_G(s)$ **then**
3:     send packet with $\texttt{header} = (\texttt{local}, s, t)$ using $\texttt{port}_s(t)$ stored in $\texttt{tbl}(s)$
4:   **else if** u knows $SP(s,t)$ /* due to handshake */ **then**
5:     send packet with $\texttt{header} = (\texttt{direct}, s, t, 0, SP(s,t))$ using port $SP(s,t)[0]$
6:   **else**
7:     send packet with $\texttt{header} = (\texttt{toLandmark}, s, \texttt{addr}(t))$ using $\texttt{port}_s(\ell(t))$ stored in $\texttt{tbl}(s)$
8:   **end if**
9:   **exit**
10: **end if**
11: /* $u \neq s$ */
12: **if** $u = \texttt{header}.t$ **then**
13:   **exit** as the packet arrived.
14: **end if**
15: **if** $\texttt{header}.route = \texttt{toLandmark}$ **then**
16:   **if** $u = \texttt{header}.addr.\ell(t)$ **then**
17:     $\texttt{header}.route \leftarrow \texttt{fromLandmark}$; $\texttt{header}.pos \leftarrow 0$; $\texttt{header}.SP \leftarrow \texttt{header}.addr.SP(\ell(t), t)$;
18:     forward packet with the new `header` using port $\texttt{header}.SP[0]$
19:   **else**
20:     forward the packet to $\texttt{port}_u(\texttt{header}.addr.\ell(t))$ stored in $\texttt{tbl}(u)$
21:   **end if**
22: **else if** $\texttt{header}.route \in \{\texttt{fromLandmark}, \texttt{direct}\}$ **then**
23:   $\texttt{header}.pos \leftarrow \texttt{header}.pos + 1$
24:   forward the packet using port $\texttt{header}.SP[\texttt{header}.pos]$
25: **else if** $\texttt{header}.route = \texttt{local}$ **then**
26:   forward the packet using $\texttt{port}_u(\texttt{header}.t)$ stored in $\texttt{tbl}(u)$
27: **end if**

An additional handshake protocol (Algorithm 2) handles the special case when $t \notin B_G(s)$ but $s \in B_G(t)$. In this case, the basic LandmarkBallRouting scheme only achieves worst-case stretch 5 instead of 3. However, t knows the reverse path from t to s. Since the graph is undirected, t can send a special `handshake` message back to s (Line 2), and each node along the path encodes the reverse port number such that, in the end, s knows the shortest path from s to t (Lines 3–10). For simplicity of exposition we use the reasonable assumption [3] that node

u knows the port q on which the message is received. If this assumption does not hold, our handshake protocol can be adapted accordingly as follows. In the routing table of a node u, for all $v \in B_G(u) \cup \mathsf{core}_{\gamma'}(G)$, we also store a $\texttt{rev-port}_u(v) = \texttt{port}_w(u)$, where w is the first node on the path from u to v. Then, when forwarding the `handshake` message from t to s, every node u on the path (including t) prepends $\texttt{rev-port}_u(s)$ to the SP in the header. This increases the routing table size by at most $\lceil \log_2 n \rceil$ bits per entry. Note that, in Algorithm 2, we also include the case of $s \in \mathsf{core}_{\gamma'}(G)$ (see Line 1), in which case the stretch is improved from 3 to 1.

Algorithm 2 Handshake protocol on node u upon the receipt of a packet from a port q with header `header`.

```
1: if header.route = fromLandmark and u = header.t and header.s ∈ B_G(u) ∪ core_γ'(G) then
2:     send packet with header = (handshake, u, header.s, Nil) using port_u(header.s) stored in tbl(u).
3: else if header.route = handshake then
4:     header.SP = q · header.SP /* prepend the port q as part of the reverse path */
5:     if header.t = u /* reach handshake destination */ then
6:         store SP(u, header.s) = header.SP locally for later use (see Line 4 of LandmarkBallRouting.)
7:     else
8:         forward packet with the new header to port_u(header.t) stored in tbl(u).
9:     end if
10: end if
```

The performance of Algorithms 1 and 2 is evaluated in the following theorem, which is proven in the next section.

Theorem 1. LandmarkBallRouting *together with the handshake protocol is a routing scheme with the following properties: (1) the worst-case stretch is* 5 *without handshaking, (2) the worst-case stretch is* 3 *after handshaking, and (3) every routing decision takes constant time. In addition, for random graphs sampled from* $\mathbf{RPLG}(n, \tau)$*, the following properties hold: (4) the expected maximum table size is* $O(n^\gamma \log n)$ *bits; this bound also holds with probability at least* $1 - 1/n$*, (5) address length and message header size are* $O(\log n \log \log n)$ *bits with probability* $1 - o(1)$*, and (6) addresses and routing tables can be generated efficiently in expected time* $O(n^{1+\gamma} \log n)$ *and this bound also holds with probability at least* $1 - 1/n$.

4 Analysis

In this section, we analyze the performance of LandmarkBallRouting for random power-law graphs.

4.1 Stretch

The proofs use the triangle inequality as in [14, 36].

Lemma 1. LandmarkBallRouting *has worst-case stretch* 5*. After handshaking with stretch* 5*,* LandmarkBallRouting *has worst-case stretch* 3.

Proof. By the triangle inequality [14], it is easy to verify the worst-case stretch 3 after handshaking. Before handshaking, the worse-case stretch happens when $t \notin B_G(s)$ and $s \in B_G(t)$. It holds that $d(s,t) \geq d(s, \ell(s))$. The radius of t's ball is at most $d(t, \ell(t)) \leq d(t, \ell(s)) \leq d(\ell(s), s) + d(s,t)$. Also, the distance from s to t's landmark is at most $d(s, \ell(t)) \leq d(s,t) + d(t, \ell(t))$. This results in a total path length of at most

$$d(s, \ell(t)) + d(\ell(t), t) \leq d(s,t) + 2d(t, \ell(t)) \leq d(s,t) + 2(d(\ell(s), s) + d(s,t)) \leq 5d(s,t).$$

□

4.2 Random Power-Law Graphs and their Cores and Balls

We first prove some properties of the adapted random power-law graph model. Let G be a random graph sampled from $\mathbf{RPLG}(n,\tau)$. For a set of nodes S, define its *volume* $Vol(S)$ as the sum of all its nodes' w_i, that is, $Vol(S) := \sum_{v_i \in S} w_i$. We abbreviate $Vol(G) = Vol(V(G))$. Note that $Vol(G) = 1/\rho$. Let $vol(S)$ denote the sum of the nodes' degrees in the actual graph G, $vol(S) := \sum_{v_i \in S} \deg_G(v_i)$. The following lemma proves that $Vol(G)$ is linear in n.

Lemma 2. *Let G be a random graph sampled from $\mathbf{RPLG}(n,\tau)$. The volume $Vol(G)$ satisfies*

$$n < Vol(G) \le \frac{\tau-1}{\tau-2} n.$$

Proof. Lower bound: it holds that $\sum_k w_k > n$, as $\forall k < n : w_k > 1$ and $w_n = 1$.
Upper bound: it holds that

$$Vol(G) = \sum_{k=1}^{n} w_k < w_1 + \int_1^n \left(\frac{n}{x}\right)^{1/(\tau-1)} \mathrm{d}x \le \frac{\tau-1}{\tau-2} n.$$

□

In the following, we show concentration results for the actual degree of a vertex and for the volume of a set of vertices in the adapted $\mathbf{RPLG}(n,\tau)$ model. We also restate the corresponding results in the original **FDRG** model.

Lemma 3 ([11, Lemma 5.6], generalized from [29, Theorem 2.7]). *For a random graph sampled from $\mathbf{FDRG}(\boldsymbol{w})$, the random variable D_i measuring the degree of vertex v_i is concentrated around its expectation w_i as follows:*

$$\Pr[D_i > w_i - c\sqrt{w_i}] \ge 1 - e^{-c^2/2} \tag{2}$$

$$\Pr[D_i < w_i + c\sqrt{w_i}] \ge 1 - e^{-\frac{c^2}{2(1+c/(3\sqrt{w_i}))}} \tag{3}$$

Lemma 4 ([11, Lemma 5.9]). *For a random graph sampled from $\mathbf{FDRG}(\boldsymbol{w})$, for a subset of vertices S and for all $0 < c \le \sqrt{Vol(S)}$,*

$$\Pr[|vol(S) - Vol(S)| < c\sqrt{Vol(S)}] \ge 1 - 2e^{-c^2/6}.$$

Lemma 5. *Let $n \ge 4^{\frac{\tau-1}{(\tau-2)^2}}$. For a random graph sampled from $\mathbf{RPLG}(n,\tau)$, if $w_i \ge 32 \ln n$, for vertex v_i, the degree D_i satisfies the following: $\Pr[w_i/4 \le D_i \le 3w_i] > 1 - 2/n^4$.*

Proof. Recall that $\rho = 1/Vol(G) < 1/n$ (by Lemma 2). For $1 \le i \le n$, let $h(i) \in [1,n]$ denote the smallest integer such that $\rho w_{h(i)} w_i \le 1$. Consider $h(1)$. Since $\rho w_1 (\frac{n}{n^{3-\tau}})^{1/(\tau-1)} \le 1$, we have that $h(1) \le \lceil n^{3-\tau} \rceil$. Therefore, for all $1 \le i \le n$, $h(i) \le h(1) \le \lceil n^{3-\tau} \rceil$.

We split the degree D_i into two parts: the contribution by edges to nodes v_j with $j < h(i)$ and the contribution stemming from edges to nodes v_j with $j \ge h(i)$. When $h(i) \ge 1$, there are at least $h(i) - 1$ edges to nodes v_j with $j \le h(i)$. Now consider the edges between v_i and v_j for $j \ge h(i)$. Since the sequence $\boldsymbol{w}$ is monotonically decreasing,

$$\begin{aligned}
\sum_{i=h(i)}^{n} w_i &\ge \int_{n^{3-\tau}+1}^{n} (n/x)^{1/(\tau-1)} \mathrm{d}x \\
&\ge \frac{\tau-1}{\tau-2}\left(n - n^{1/(\tau-1)} 2^{\frac{\tau-2}{\tau-1}} n^{\frac{\tau-2}{\tau-1}(3-\tau)}\right) \quad \text{/* use condition: } (n^{3-\tau} \ge 1) \text{ */} \\
&\ge \frac{\tau-1}{2(\tau-2)} n. \quad \text{/* use condition: } n \ge 4^{\frac{\tau-1}{(\tau-2)^2}} \text{ */}
\end{aligned}$$

Recall that $\rho = 1/\sum_{i=1}^n w_i \geq \frac{\tau-2}{n(\tau-1)}$ by Lemma 2. Let D_i' be the random variable denoting the number of edges from v_i to v_j with $j \geq h(i)$ in a random graph. Thus, $E[D_i'] = \mu = \rho w_i \sum_{j=h(i)}^n w_i \geq w_i/2 \geq 16 \ln n$. Also $\mu \leq w_i$. Since there are no deterministic edges in this case, the random variable D_i' can be bounded using Lemma 3:

$$\mathsf{Pr}[D_i' > \mu/2] \geq 1 - e^{-\mu/4} \geq 1 - 1/n^4,$$
$$\mathsf{Pr}[D_i' < 2\mu] \geq 1 - e^{-3\mu/8} \geq 1 - 1/n^4.$$

If $h(i) = 1$, the lemma follows directly. If $h(i) > 1$, we have $D_i \leq D_i' + h(i) - 1$. Notice that $\rho w_i (n/w_i)^{1/(\tau-1)} \leq 1$, which implies that $h(i) \leq \lceil w_i \rceil \leq w_i + 1$. Therefore,

$$\mathsf{Pr}[w_i/4 \leq \mu/2 \leq D_i \leq 3w_i] \leq 1 - 2/n^4.$$

□

Lemma 6. *Let G be a random graph sampled from $\mathbf{RPLG}(n,\tau)$. For a subset of vertices S satisfying $Vol(S) \geq 192 \ln n$, it holds with probability at least $1-2/n^3$ that $Vol(S)/8 \leq vol(S) \leq 4\,Vol(S)$.*

Proof. We split S into two parts. Nodes v_i with small w_i, $S_1 := \{v_i \in S : w_i < 32 \ln n\}$, and nodes v_i with large w_i, $S_2 = S \setminus S_1$. By Lemma 5, $\mathsf{Pr}[Vol(S_2)/4 \leq vol(S_2) \leq 3\,Vol(S_2)] \geq 1 - 2\,|S_2|\,/n^4$.

As for each vertex $v_i \in S_1$, $w_i < 32 \ln n$, we can apply Lemma 4 to S_1, since no deterministic edges are attached to S_1. Therefore, if $Vol(S_1) \geq 96 \ln n$, by Lemma 4, $\mathsf{Pr}[\,Vol(S_1)/2 \leq vol(S_1) \leq 2\,Vol(S_1)/3] \geq 1 - 2/n^4$. Therefore, the statement holds with probability at least $1 - 2(|S_2| + 1)/n^4 \geq 1 - 2/n^3$.

If $Vol(S_1) < 96 \ln n$, we have $Vol(S_2) \geq Vol(S)/2 \geq 96 \ln n$. Nevertheless, we can still apply Lemma 4 to bound $vol(S_1)$ from above as $\mathsf{Pr}[vol(S_1) < \frac{3}{2} \cdot 96 \ln n \leq \frac{3}{4}\,Vol(S)] \geq 1 - 2/n^4$. In this case, since $\mathsf{Pr}[\,Vol(S)/8 \leq Vol(S_2)/4 \leq vol(S_2) \leq 3\,Vol(S_2)] \geq 1 - 2\,|S_2|\,/n^4$, the statement also holds with probability at least $1 - 2/n^3$.

□

Corollary 1. *The number of edges of a random graph sampled from $\mathbf{RPLG}(n,\tau)$ is at most $vol(G)/2 \leq \frac{4(\tau-1)}{\tau-2} n$ with probability at least $1 - 1/n^2$.*

There is an edge between two nodes v_i, v_j with probability proportional to w_i and w_j. This is generalized for sets of nodes $S, T \subseteq V(G)$ in the following and holds for both $\mathbf{FDRG}(\boldsymbol{w})$ and $\mathbf{RPLG}(n,\tau)$.

Lemma 7 ([10, Lemma 3.3], proof in [28, Lemma 9]). *For any two disjoint subsets S and T with $Vol(S) \cdot Vol(T) > c \cdot Vol(G)$, we have*

$$\mathsf{Pr}[d(S,T) > 1] = \prod_{v_i \in S, v_j \in T} \max\{0, (1 - w_i w_j / Vol(G))\} \leq e^{-c}.$$

4.3 Core size.

To compute the size of $\mathsf{core}_{\gamma'}(\boldsymbol{w})$, we solve the inequality $w_k > n^{\gamma'}$ and obtain k.

$$w_k = \left(\frac{n}{k}\right)^{\frac{1}{\tau-1}} > n^{\gamma'} \Leftrightarrow k^{-\frac{1}{\tau-1}} > n^{\gamma' - \frac{1}{\tau-1}}$$
$$\Leftrightarrow k < n^{(1-\tau)(\gamma' - \frac{1}{\tau-1})} = n^{\gamma'(1-\tau)+1}$$

As $\gamma' = \frac{1-\gamma}{\tau-1}$, we have $\left|\mathsf{core}_{\gamma'}(\boldsymbol{w})\right| = \lceil n^{\gamma'(1-\tau)+1}\rceil - 1 = \lceil n^{\gamma}\rceil - 1$.

Even if the same degree threshold $n^{\gamma'}$ is used for $\mathsf{core}_{\gamma'}(\boldsymbol{w})$ and $\mathsf{core}_{\gamma'}(G)$, the two sets of nodes may differ. For a slightly smaller degree threshold $n^{\gamma'}/4$ (as in Definition 2), the core of the actual graph contains $\mathsf{core}_{\gamma'}(\boldsymbol{w})$ with high probability (apply Lemma 5).

Lemma 8. *Let G be a random graph sampled from $\mathbf{RPLG}(n,\tau)$. With probability at least $1-1/n^2$ it holds that $\mathsf{core}_{\gamma'}(\boldsymbol{w}) = \{v_i : w_i > n^{\gamma'}\} \subseteq \{v_i : \deg(v_i) > n^{\gamma'}/4\} = \mathsf{core}_{\gamma'}(G)$.*

Proof. Let v_i be a vertex in $\mathsf{core}_{\gamma'}(\boldsymbol{w})$. By Lemma 5, $D_i \geq n^{\gamma'}/4$ with probability at least $1-2/n^4$. This holds for all $j \leq i$. Therefore, by union bound, the probability that $\mathsf{core}_{\gamma'}(\boldsymbol{w}) \subseteq \{v_i : \deg(v_i) > n^{\gamma'}/4\}$ is at least $1-1/n^2$. □

Lemma 9. *Let G be a random graph sampled from $\mathbf{RPLG}(n,\tau)$. With probability at least $1-1/n^2$, $\left|\mathsf{core}_{\gamma'}(G)\right| = \Theta(n^{\gamma})$.*

Proof. Since $\mathsf{core}_{\gamma'}(G)$ contains $\mathsf{core}_{\gamma'}(\boldsymbol{w})$ with high probability $(1-1/n^2)$, its size is at least n^{γ} with high probability.

Let $i = 144n^{\gamma}$. By Lemma 5, $D_i \leq 3w_i < n^{\gamma'}/4$ with probability at least $1-2/n^4$. This holds for all $j \in (i,n]$. By union bound, $\mathsf{core}_{\gamma'}(G)$ does not contain any vertex v_j for $i \leq j \leq n$, with probability at least $1-1/n^2$, which implies $\left|\mathsf{core}_{\gamma'}(G)\right| \leq 144n^{\gamma}$ with probability at least $1-1/n^2$. □

4.4 Ball sizes.

Let G be a random graph sampled from random power-law graph. Recall that a ball is defined by

$$B_G(u) = \{v \in V(G) \,:\, d(u,v) < \min_{v' \in \mathsf{core}_{\gamma'}(G)} d(u,v')\}.$$

Lemma 10. *Let $\beta = \gamma'(\tau-2) + \frac{(2\tau-3)\varepsilon}{\tau-1}$ be a constant. Assume Equation (1) is satisfied. For a random graph G sampled from $\mathbf{RPLG}(n,\tau)$, with probability at least $1-3/n^2$, it holds that for all $u \in V(G)$,*

$$\begin{aligned} |B_G(u)| &= \left|\{u' \in V(G) : d(u,u') < d(u, \mathsf{core}_{\gamma'}(\boldsymbol{w}))\}\right| = O(n^{\beta}), \\ |E(B_G(u))| &= O(n^{\beta}\log n), \end{aligned}$$

where $E(B_G(u))$ is the set of internal edges among vertices in $B_G(u)$.

Since for $\mathbf{RPLG}(n,\tau)$ the edges are independent, in our analysis, the existence of every edge in random graph G is only determined when it is needed, and before that it is treated as a probability distribution as defined in our random graph model. We call the determination of the existence of an edge according to its probability distribution *revealing* the edge.

For a given vertex $u \in V(G)$, we define a sequence of balls $(B_0 = \{u\}, B_1, B_2, \ldots)$ as follows: Let $V' = V \setminus \mathsf{core}_{\gamma'}(\boldsymbol{w})$. Now define $B_i = \{v \,:\, d_G(u,v) \leq i\}$. We also define the circles $C_i = B_i \setminus B_{i-1}$ for $i \geq 0$ with $B_{-1} = \emptyset$. Let E_i be the number of edges between C_i and $C_i \cup C_{i+1}$. We first give a concentration result on E_i.

Lemma 11. *For circle C_i, the following holds with probability at least $1-2/n^3$:*

$$\begin{aligned} &\textit{If } \mathit{Vol}(C_i) < 192\ln n, \textit{ then } E_i \leq 4\cdot 192\ln n, \textit{and} \\ &\textit{if } \mathit{Vol}(C_i) \geq 192\ln n, \textit{ then } E_i \leq 4\,\mathit{Vol}(C_i). \end{aligned}$$

Proof. For our analysis, we assume that the edges of the random graph are revealed in consecutive steps as follows: in step i with $i \geq 0$, edges from C_i to $V' \setminus B_{i-1}$ are revealed and circle C_{i+1} is formed. In other words, when discovering C_i, the edges between C_i and $V'' = V' \setminus B_{i-1}$ have not been revealed yet.

In particular, E_i measures the number of edges between C_i and V'' under the condition that we know all edges adjacent to B_{i-1}. We can define another random graph G' on the vertex set V'', such that the edge between two vertices in V'' is sampled with the same probability as in $\mathbf{RPLG}(n, \tau)$. Clearly, E_i and $vol_{G'}(C_i)$ have the same distribution, where $vol_{G'}(C_i)$ denotes the number of edges adjacent to C_i in G'.

Let $vol(C_i)$ denote the random variable measuring the number of edges adjacent to C_i in the original model **FDRG**. $vol_{G'}(C_i)$ is *stochastically dominated* by $vol(C_i)$. Hence, the lemma directly follows since it applies to $vol(C_i)$ by Lemma 6. $\square$

Since there are at most n circles, Lemma 11 holds for all circles with probability at least $1 - 2/n^2$. We are now ready to prove Lemma 10.

Proof (of Lemma 10).
Let k be the smallest integer such that $Vol(B_k) \geq n^\beta$. We have the conditions $Vol(B_k) \geq n^\beta$, $Vol(\mathsf{core}_{\gamma'}(\boldsymbol{w})) \geq |\mathsf{core}_{\gamma'}(\boldsymbol{w})|\, n^{\gamma'} = n^{\gamma+\gamma'}$, and $Vol(G) \leq \frac{\tau-1}{\tau-2}n$ (Lemma 2). From Equation (1), $n^{\beta-\gamma'(\tau-2)} > 2\frac{\tau-1}{\tau-2}\ln n$. Since the edges between B_k and $\mathsf{core}_{\gamma'}(\boldsymbol{w})$ have not been revealed, Lemma 7 can be applied. Due to Lemma 7, there is an edge between B_k and $\mathsf{core}_{\gamma'}(\boldsymbol{w})$ with probability at least $1-1/n^2$. Recall that $\mathsf{core}_{\gamma'}(\boldsymbol{w}) \subseteq \mathsf{core}_{\gamma'}(G)$ with probability at least $1-1/n^2$ by Lemma 8. Hence $B_G(u) \subseteq B_k$ with probability at least $1 - 2/n^2$.

In the following, we bound the size of B_k. Lemma 11 holds for all circles with high probability. In our case, $Vol(C_{k-1}) \leq Vol(B_{k-1}) < n^\beta$. By Lemma 11, $|C_k| \leq E_{k-1} \leq 4n^\beta$ with probability at least $1 - 1/n^2$. Then, $|B_k| = |B_{k-1}| + |C_k| \leq Vol(B_{k-1}) + |C_k| \leq 5n^\beta$.

Since $B_G(u) \subseteq B_k$ with probability at least $1-2/n^2$, we have $|E(B_G(u))| = O(vol(B_{k-1}(u))) = O\left(\sum_{i=0}^{k-1} E_i\right)$, with probability at least $1 - 2/n^2$.

By Lemma 11, with probability at least $1 - 1/n^2$, $E_i \leq 4 \cdot 192 \ln n + 4\,Vol(C_i)$ for all i. Since $k \leq n^\beta$, with probability at least $1 - 3/n^2$,

$$|E(B_G(u)| = O\left(\sum_{i=0}^{k-1} E_i\right) = O(4 \cdot 192 n^\beta \ln n + 4\,Vol(B_{k-1})) = O(n^\beta \log n).$$

$\square$

4.5 Table Sizes and Computations

The core $\mathsf{core}_{\gamma'}(G)$ has size $\Theta(n^\gamma)$ with probability at least $1 - 1/n^2$ (Lemma 9) and all balls $B_G(u)$ have size $O(n^\gamma)$ with probability at least $1 - 3/n^2$ (Lemma 10). Therefore, we have the following result.

Lemma 12. *For a random graph G sampled from $\mathbf{RPLG}(n, \tau)$, for all $u \in V(G)$, the expected table size is at most*

$$|\mathtt{tbl}(u)| = O(n^\gamma)$$

and all tables can be generated in expected time at most $O(n^{1+\gamma}\log n)$. These bounds also hold with probability at least $1 - 1/n$.

Proof. Note that each entry of $\mathtt{tbl}(u)$ has $O(\log n)$ bits. Thus the total table size per node is $O(n^\gamma \log n)$ bits.

Our algorithm is deterministic. The expected time (space) complexity is the average running time (space) of our algorithm over all graphs from the random graph distribution $\mathbf{RPLG}(n, \tau)$.

Given a graph G with n nodes and m edges, our algorithm computes the core $\mathsf{core}_{\gamma'}(G)$ of G with time complexity $O(m + n \log n)$. It runs a complete breadth-first search for each node of the core in time $O(m)$. Let $B_G(u)$ be the ball computed in our algorithm for vertex u. Let $T(B_G(u))$ denote the time to compute $B_G(u)$. Therefore, the time complexity TC and space complexity SC of our algorithm are at most

$$TC(G) = O\left(m \cdot \left|\mathsf{core}_{\gamma'}(G)\right| + \sum_{v \in V(G)} T(B_G(u))\right), \tag{4}$$

$$SC(G) = O\left(n \cdot \left|\mathsf{core}_{\gamma'}(G)\right| + \sum_{v \in V(G)} |B_G(u)|\right). \tag{5}$$

We now know that with probability at least $1 - 5/n^2$, all of the following conditions are true: (1) $m = \Theta(n)$ (Corollary 1); (2) $|\mathsf{core}_{\gamma'}(G)| = \Theta(n^\gamma)$ (Lemma 9); (3) $|B_G(u)| = O(n^\beta)$ for all vertices u (Lemma 10); (4) $T(B_G(u)) = O(n^\beta \log n)$ for all vertices u (Lemma 10). Therefore, from Equations (4) and (5), we know that with probability at least $1-5/n^2$, the space complexity of our algorithm is $O(n^{1+\gamma} + n^{1+\beta})$ and the time complexity is $O(n^{1+\gamma} + n^{1+\beta} \log n)$.

Finally, we fix the parameters to obtain a balanced scheme. In a balanced scheme, the core size and the expected ball sizes are asymptotically equivalent, that is, $\beta = \gamma$. Together with $\beta = \gamma'(\tau-2)+\frac{(2\tau-3)\varepsilon}{\tau-1}$ and $\gamma' = \frac{1-\gamma}{\tau-1}$, we have $\gamma = \frac{\tau-2}{2\tau-3}+\varepsilon$. Therefore, assuming that Equation (1) is satisfied, the space requirement per node is $O(n^\gamma \log n)$ bits and the preprocessing time is bounded by $O(n^{1+\gamma} \log n)$, which holds with probability at least $1 - 1/n$. □

4.6 Address Lengths

We now bound the number of bits for the address of each vertex. For one vertex u, its address contains the encoding of the shortest path $SP(u, \ell(u))$ from u to its landmark $\ell(u)$. We need to bound the diameter of a random power-law graph and the diameter of its core. The proofs in [10] on diameters can be carried over to our adapted model.

Lemma 13 (Chung and Lu [10, Claim 4.4]). *For a random graph sampled from $\mathbf{RPLG}(n, \tau)$, with probability at least $1 - o(1)$, the diameter of its largest connected component is $\Theta(\log n)$.*

By Lemma 13, the length of $SP(u, \ell(u))$ is at most $O(\log n)$ asymptotically almost surely. Therefore, $SP(s,t)$ can be encoded with $O(\log^2 n)$ bits. This bound can be improved to $O(\log n \cdot \log\log n)$, as proven in the following lemma.

Lemma 14. *For a random graph G sampled from $\mathbf{RPLG}(n, \tau)$, with probability at least $1-o(1)$, it holds that for all $s, t \in V(G)$, $SP(s,t)$ can be encoded with $O(\log n \log\log n)$ bits.*

The proof is split into several claims from [10]. We first extend the core.

Definition 4. *The* extended core *of a random graph from $\mathbf{RPLG}(n, \tau)$ contains all nodes v_i with w_i at least $n^{1/\log\log n}$, that is, $\mathsf{core}^+(\boldsymbol{w}) = \{v_i \in V : w_i \geq n^{1/\log\log n}\}$.*

Note that, as τ is a constant, $1/\log\log n \leq \gamma'$ for large enough n, and thus $\mathsf{core}^+(\boldsymbol{w}) \supseteq \mathsf{core}_{\gamma'}(\boldsymbol{w})$. The following lemma constitutes a bound for the diameter of the core. This is from the fact that the *extended core* "contains" a dense Erdős-Rényi [17] random graph.

Lemma 15 (Chung and Lu [10, Claim 4.1]). *Let G be a random graph sampled from* $\mathbf{RPLG}(n,\tau)$*. The diameter of the subgraph induced by* $\mathsf{core}^+(\boldsymbol{w})$ *in G is $O(\log\log n)$ with probability at least $1-1/n$.*

The next lemma proves that a vertex v_i with large enough w_i is close to the *extended core*.

Lemma 16 (Chung and Lu [10, Claim 4.2]). *Let G be a random graph sampled from* $\mathbf{RPLG}(n,\tau)$*. There exists a constant C, such that each vertex v_i with $w_i \geq \log^C n$ is at distance $O(\log\log n)$ from the* extended core*, with probability at least $1-1/n^2$.*

Corollary 2 (Corollary of Lemma 16). *Let G be a random graph sampled from* $\mathbf{RPLG}(n,\tau)$*. Let C be the constant in Lemma 16. With probability at least $1-1/n$, the distance between any two vertices v_i, v_j with $w_i \geq \log^C n$ and $w_j \geq \log^C n$ is $O(\log\log n)$.*

Proof (of Lemma 14). Let v_i and v_j be the first and the last vertex in $SP(s,t)$ from s to t such that w_i and w_j both are greater than $\log^C n$, where C is the constant from Lemma 16. By Corollary 2, with probability $1-1/n$, the portion of the shortest path $SP(s,t)$ between v_i and v_j has length at most $O(\log\log n)$. Therefore, this portion of the shortest path can be encoded with $O(\log n \log\log n)$ bits, with probability $1-1/n$.

For the rest of the shortest path, each node has w_i at most $\log^C n$. By Lemma 5, all such nodes have degree at most $3\log^C n$ with probability at least $1-2/n^3$. To encode the next neighbor in the shortest path, at most $O(\log\log n)$ bits are necessary. Since $SP(s,t)$ contains $O(\log n)$ nodes with probability $1-o(1)$ (Lemma 13), the rest of the shortest path can also be encoded with $O(\log n \log\log n)$ bits, with probability $1-o(1)$. □

Corollary 3. *For a random graph G sampled from* $\mathbf{RPLG}(n,\tau)$*, with probability at least $1-o(1)$, it holds that for all $u \in V(G)$, the address* $\texttt{addr}(u)$ *can be encoded with $O(\log n \log\log n)$ bits.*

5 Experiments

In this section, we experimentally demonstrate the efficiency of our scheme. We use the following datasets in our experiments.

Real-world graphs. The most important application scenario for a compact routing scheme is arguably a communication network. The router-level topology of a portion of the Internet, measured by CAIDA [13], is an undirected, unweighted graph with $190,914$ nodes and $607,610$ edges. The estimated power-law exponent (maximum likelihood method [32]) is $\hat{\tau} = 2.82$.

Random Power-Law Graphs. We extracted the largest connected component from the random power-law graphs generated by Brady and Cowen [8] (pre-generated graphs, $N = 10,000$ and $\tau \in (2,3)$, downloaded from `http://digg.cs.tufts.edu/`).

In addition, we generated graphs of 10,000 nodes with the tool BRITE [30] using the configurations for the Barabási [7] and Waxman [38] models for an Autonomous System Topology (AS) and a Router Topology (RT) — the precise configurations are listed in Section 5.1. The edge weights were ignored and the links interpreted as undirected.

Note that for all the random graphs considered, the generation process does not exactly match the $\mathbf{RPLG}(n,\tau)$.

Graph	CAIDA [13]	ASBarabasi	RTBarabasi	ASWaxman	RTWaxman
random, $p = n^{-1/2}$	929.84±95.40	204.03±25.57	208.32±22.21	221.95± 24.73	217.75± 28.00
highdeg, $\lceil n^\gamma \rceil$	173.68±55.80	32.16±41.30	44.95±58.21	139.45±142.94	130.65±131.78
Graphs [8]	$\tau = 2.1$	$\tau = 2.2$	$\tau = 2.3$	$\tau = 2.4$	$\tau = 2.5$
random, $p = n^{-1/2}$	74.90±37.96	74.94±44.78	77.49±50.56	79.74± 55.50	82.54± 60.17
highdeg, $\lceil n^\gamma \rceil$	55.20±67.48	48.50±54.57	42.20±42.94	43.28± 40.10	43.55± 38.37
Graphs [8]	$\tau = 2.6$	$\tau = 2.7$	$\tau = 2.8$	$\tau = 2.9$	
random, $p = n^{-1/2}$	86.88±69.69	85.56±71.35	84.69±73.87	76.65± 71.71	
highdeg, $\lceil n^\gamma \rceil$	45.59±39.59	50.24±46.08	56.48±56.26	46.85± 46.65	

Table 1. Table sizes: mean and standard deviation

Graph	CAIDA [13]	ASBarabasi	RTBarabasi	ASWaxman	RTWaxman
random	1.28±0.16	1.38±0.28	1.38±0.25	1.37±0.25	1.38±0.16
highdeg, $\lceil n^\gamma \rceil$	1.12±0.14	1.15±0.21	1.20±0.22	1.36±0.26	1.35±0.24
Graphs [8]	$\tau = 2.1$	$\tau = 2.2$	$\tau = 2.3$	$\tau = 2.4$	$\tau = 2.5$
random, $p = n^{-1/2}$	1.34±0.24	1.35±0.24	1.35±0.25	1.34±0.26	1.34±0.26
highdeg, $\lceil n^\gamma \rceil$	1.30±0.24	1.26±0.23	1.23±0.23	1.21±0.23	1.18±0.22
Graphs [8]	$\tau = 2.6$	$\tau = 2.7$	$\tau = 2.8$	$\tau = 2.9$	
random, $p = n^{-1/2}$	1.33±0.28	1.31±0.28	1.29±0.29	1.25±0.28	
highdeg, $\lceil n^\gamma \rceil$	1.16±0.22	1.15±0.22	1.15±0.24	1.11±0.22	

Table 2. Stretch: mean and standard deviation

Routing schemes. In the specification of our routing scheme LandmarkBallRouting, we use $n^{\gamma'}/4$ as a degree threshold (Definition 2) and obtain a core of size $\Theta(n^\gamma)$. The largest connected components of the graphs generated by Brady and Cowen [8] and the graphs generated using BRITE [30] do not contain nodes with such a high degree. Therefore, for the experiments with our routing scheme, the algorithm selects the $\lceil n^\gamma \rceil$ nodes with the highest degrees as landmarks. In practice, this might indeed be a better strategy.

We compare our high-degree selection strategy with the random selection with probability $n^{-1/2}$, which is *similar* to Thorup and Zwick [36] for $k = 2$. Recall that their scheme is not optimized for power-law graphs but works for general, weighted graphs as well. We also compare our scheme with the values obtained by Brady and Cowen [8].

Settings and results. For the graphs generated by Brady and Cowen [8], the high-degree selection and the random sampling process were executed five times for each of the ten graphs per value of τ, which gives a total of $5 \cdot 10 \cdot 9 \cdot 2 = 900$ routing scheme constructions. For each of the remaining graphs (Barabási, Waxman, CAIDA), both schemes were constructed at least 10 times. We report the table sizes (mean and standard deviation) in Table 1. For each instance, 200 random (s, t) pairs were generated and packets routed. The stretch (the length of the route divided by the length of a shortest path) is reported in Table 2.

In our experiments, the strategy of selecting few high-degree nodes as landmarks always produces significantly smaller routing tables compared to a large number of landmarks selected at random. The best results are achieved for the graphs stemming from the Barabási model, for which the high-degree-based tables are roughly 5 times smaller than their random-based counterpart. The average table size for the randomly selected landmarks is close to $\sqrt{n}$, which means that most balls are actually (almost) empty. As predicted by our analysis, this indicates that, for power-law graphs, the optimal balance for randomly selected landmarks may be smaller than $O(\sqrt{n})$.

The average stretch is surprisingly consistent among different datasets. Even though there are fewer landmarks, the average stretch is better if high-degree nodes are selected as landmarks. Brady and Cowen [8] claim average stretch 1.18–1.25 for the scheme by Thorup and Zwick [36]. Our experiments do not confirm this claim: randomly selected nodes (similar to TZ) did not achieve this stretch. Brady and Cowen also claim average stretch 1.11–1.22 for their scheme and small values for $\tau \in \{2.1, 2.2, 2.3\}$. Our scheme, except for the graphs of the Waxman model and for small values of $\tau \leq 2.2$, also achieves these average stretch values. The worst-case stretch is difficult to compare as our scheme has a (non-experimental) worst-case *multiplicative* stretch and the scheme by Brady and Cowen has an experimental worst-case *additive* stretch. Brady and Cowen conclude from their topology experiments that, for graphs up to $40,000$ nodes, their scheme has a worst-case additive stretch of 10 while maintaining $O(\log^2 n)$-bit tables per node. For nodes 'close' to each other (distance less than 5), the multiplicative stretch of 3 yields better stretch guarantees. For nodes 'far' from each other (distance at least 5), the additive stretch of 10 yields better stretch guarantees. In power-law graphs, most distances are short, the typical distance being $O(\log \log n)$ [10].

The high-degree nodes in the power-law graphs of the Waxman model have only very few edges: the highest degree is only 20. Furthermore, as $\lceil n^\gamma \rceil = 3$, the core is really small and so is the cumulative degree. Compared to the other power-law graphs, the high-degree selection strategy does not produce huge benefits but it still outperforms random selection. In practice, one might add high-degree nodes to the set of landmarks until a certain cumulative degree threshold (for example $\sqrt{n}$ or also a threshold value dependent on τ) is reached.

5.1 Details for the BRITE graphs used in the experiments

We provide the detailed parameters used to generate the graphs using BRITE [30], based on the Barabási [7] and Waxman [38] models. We use the prefix of AS to denote the Autonomous System topology and RT to denote the Router Topology.
Model (1 - RTWaxman): 10000 1000 100 1 2 0.15 0.2 1 1 10.0 1024.0
Model (2 - RTBarabasi): 10000 1000 100 1 2 1 10.0 1024.0
Model (3 - ASWaxman): 10000 1000 100 1 2 0.15 0.2 1 1 10.0 1024.0
Model (4 - ASBarabasi): 10000 1000 100 1 2 1 10.0 1024.0

The resulting graphs have the following numbers of nodes and edges, and the corresponding power-law exponent $\hat{\tau}$, estimated using [32].

Graph	Nodes	Edges	$\hat{\tau}$
ASWaxman	10,000	20,000	2.806
RTWaxman	10,000	20,000	2.806
ASBarabasi	10,000	19,997	2.893
RTBarabasi	10,000	19,997	2.892

6 Approximate Distance Oracle

Dijkstra's algorithm [15] finds a shortest path in any graph with non-negative edge weights in time $O(n \log n + m)$, where n and m denote the number of nodes and edges respectively. For applications such as navigation software exploring huge maps or for social networking sites, this query time is not practical. Instead, the graph is *preprocessed* and a special data structure allows for efficient *queries*. One way to prepare for queries is to precompute all shortest paths using an All-Pairs Shortest Path algorithm [9] and to read a shortest path from a distance table. Time and memory constraints, however, render this approach impractical. Instead of running a cubic-time algorithm and using quadratic storage, we want to efficiently preprocess a graph to allow for fast distance queries. However, for general (directed) graphs with n vertices, $\Omega(n^2)$ space is necessary to return the shortest distance. This calls for an approximation method. Approximate distance oracles address the trade-off between approximation ratio, space, and preprocessing and query time, and can thus be interpreted as a generalization of the All-Pairs (Approximate) Shortest Path problem.

Thorup and Zwick [37] provide a stretch-$2k-1$ oracle of size $\tilde{O}(kn^{1+1/k})$, which can be constructed in time $O(kmn^{1/k})$. Assuming a girth conjecture by Erdős, stretch and size are basically tight.

To the best of our knowledge, there is no distance oracle for power-law graphs so far. We prove the following:

Theorem 2. *Let $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$ be a constant. Assume Equation (1) is satisfied. For random power-law graphs from $\mathbf{RPLG}(n,\tau)$ (Definition 1), there exists a preprocessing algorithm that runs in expected time $O(n^{1+\gamma}\log n)$ and creates a distance oracle of expected size $O(n^{1+\gamma})$. These bounds also hold with probability at least $1-1/n$. After preprocessing, approximate distance queries can be answered in $O(1)$ time with stretch at most 3.*

We propose a modification of the distance oracle by Thorup and Zwick [37, Fig. 5] for $k=2$, which guarantees stretch 3. The main idea of the scheme by Thorup and Zwick for $k=2$ is the following: in the preprocessing step, given a graph $G=(V,E)$, (1) each node $v \in V$ is chosen as a *landmark* independently at random with probability $n^{-1/2}$. The expected number of landmarks is $\sqrt{n}$. (2) For each node $u \in V$, find its nearest landmark $\ell(u)$ and compute the distances from u to all landmarks. To guarantee optimal stretch for short distance queries, (3) for every node $u \in V$ a local *ball* $B_G(u) = \{u' \in V(G) : d(u,u') < d(u,\ell(u))\}$ is computed, including all nodes with distance strictly less than the distance to the landmarks. The result of the distance query $d(s,t)$ is exact if $s \in B(t)$ or $t \in B(s)$ and otherwise stretch 3 is guaranteed [14]. Since the set of landmarks consists of a random sample, the expected ball size is $O(\sqrt{n})$, which is equal to the number of landmarks. This is the optimal balance for general graphs.

For power-law graphs a *better* balance is possible. Using high-degree nodes as landmarks is a natural heuristic. We can select fewer landmarks and obtain smaller sized balls than [37, Fig. 5] at the same time. Details for the preprocessing step are listed in Algorithm 3.

Algorithm 3 Preprocess($G=(V,E),\gamma'$)

```
compute core ← {v ∈ V : deg(v) > n^{γ'}/4}
for each v ∈ core do
    run breadth-first search from v in G
    for each node u ≠ v, store d(u,v) and set port_u(v) to be the penultimum node on the shortest path.
end for
for each u ∈ V do
    compute and store B_G(u) (including distances)
    for each v ∈ B_G(u) set port_u(v) to be the first node on the shortest path to v.
end for
```

Lemma 17. *Let $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$ be a constant. Assume Equation (1) is satisfied. For random power-law graphs $\mathbf{RPLG}(n,\tau)$, Algorithm 3 runs in expected time $O(n^{1+\gamma}\log n)$ and creates a distance oracle of expected size $O(n^{1+\gamma})$. These bounds also hold with probability at least $1-1/n$.*

Proof. The analysis of the compact routing scheme can be applied directly (Lemma 10 and proof of Lemma 12). □

Algorithm 4 distance(s, t)

```
if s ∈ B(t) or t ∈ B(s) then
  return local distance d(s, t) from tbl(s) or tbl(t).
else
  return d(s, ℓ(t)) + d(ℓ(t), t)
end if
```

The query algorithm is the same as in [37] for $k = 2$, see Algorithm 4.

Lemma 18. *Algorithm 4 runs in time $O(1)$ and achieves stretch 3.*

Proof. Stretch and time bounds from [37] apply. □

Theorem 2 is immediate from Lemmas 17 and 18.

7 Conclusion

Our analysis provides theoretical justification that high-degree nodes in power-law graphs are indeed very important for finding shortest paths in such networks, and thus are effective in improving the performance of shortest-path-related computations. With the ubiquity of power-law networks, our result suggests that, when designing network algorithms, optimizing for power-law graphs rather than dealing with general graphs, may lead to significantly better algorithm performance in real-world networks.

Perhaps the most intriguing question is whether even polylogarithmic tables would suffice to route with small stretch in power-law graphs. It also remains open whether the scheme by Thorup and Zwick for general k can be optimized for power-law graphs and whether similar techniques can be applied to the name-independent scheme by Abraham et al. [5]. An average-case analysis of the actual scheme by Thorup and Zwick would be interesting as well as a rigorous analysis of the scheme by Brady and Cowen [8]. Furthermore, the analysis for other models of random power-law graphs is an interesting open problem.

Acknowledgments. The second author thanks Mikkel Thorup for helpful comments and interesting discussions.

References

1. I. Abraham, C. Gavoille, A. V. Goldberg, and D. Malkhi. Routing in networks with low doubling dimension. In *Proceedings of the 26th International Conference on Distributed Computing Systems*, 2006.
2. I. Abraham, C. Gavoille, and D. Malkhi. Compact routing for graphs excluding a fixed minor. In *DISC*, pages 442–456, 2005.
3. I. Abraham, C. Gavoille, and D. Malkhi. On space-stretch trade-offs: lower bounds. In *SPAA*, pages 207–216, 2006.
4. I. Abraham, C. Gavoille, and D. Malkhi. On space-stretch trade-offs: upper bounds. In *SPAA*, pages 217–224, 2006.
5. I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup. Compact name-independent routing with minimum stretch. *ACM Transactions on Algorithms*, 4(3), 2008.
6. W. Aiello, F. R. K. Chung, and L. Lu. A random graph model for massive graphs. In *STOC*, pages 171–180, 2000.
7. A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
8. A. Brady and L. Cowen. Compact routing on power law graphs with additive stretch. In *Proc. of the 9th Workshop on Algorithm Eng. and Exper.*, pages 119–128, 2006.
9. T. M. Chan. More algorithms for all-pairs shortest paths in weighted graphs. In *STOC*, pages 590–598, 2007.
10. F. Chung and L. Lu. The average distances in random graphs with given expected degrees. *Internet Mathematics*, 99:15879–15882, 2002.
11. F. Chung and L. Lu. *Complex Graphs and Networks*. American Mathematical Society, 2006.
12. A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. arXiv:0706.1062, 2007.

13. Cooperative Association for Internet Data Analysis. CAIDA's router-level topology measurements. Online at http://www.caida.org/tools/measurement/skitter/router_topology/, file: itdk0304_rlinks_undirected.gz, 2003.
14. L. Cowen. Compact routing with minimum stretch. *J. Algorithms*, 38(1):170–183, 2001.
15. E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Math.*, 1:269–271, 1959.
16. M. Enachescu, M. Wang, and A. Goel. Reducing maximum stretch in compact routing. In *INFOCOM*, pages 336–340, 2008.
17. P. Erdős and A. Rényi. On the evolution of random graphs. *Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei*, 5:17–61, 1960.
18. M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *SIGCOMM: Proceedings of the conference on applications, technologies, architectures, and protocols for computer communication*, pages 251–262, 1999.
19. P. Fraigniaud and C. Gavoille. Routing in trees. In *ICALP*, pages 757–772, 2001.
20. C. Gavoille and N. Hanusse. Compact routing tables for graphs of bounded genus. In *ICALP*, pages 351–360, 1999.
21. C. Gavoille and S. Perennes. Memory requirements for routing in distributed networks (extended abstract). In *PODC*, pages 125–133, 1996.
22. G. Konjevod, A. W. Richa, and D. Xia. Optimal-stretch name-independent compact routing in doubling metrics. In *PODC*, pages 198–207, 2006.
23. G. Konjevod, A. W. Richa, D. Xia, and H. Yu. Compact routing with slack in low doubling dimension. In *PODC*, pages 71–80, 2007.
24. A. Korman. Improved compact routing schemes for dynamic trees. In *PODC*, pages 185–194, 2008.
25. D. V. Krioukov, K. R. Fall, and X. Yang. Compact routing on internet-like graphs. In *INFOCOM*, 2004.
26. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Random graph models for the web graph. In *FOCS*, pages 57–65, 2000.
27. H.-I. Lu. Improved compact routing tables for planar networks via orderly spanning trees. In *Proc. of the 8th Int. Computing and Combinatorics Conference*, pages 57–66, 2002.
28. L. Lu. *Probabilistic methods in massive graphs and Internet computing.* PhD thesis, University of California San Diego, 2002.
29. C. McDiarmid. *Probabilistic Methods for Algorithmic Discrete Mathematics*, volume 16 of *Algorithms and Combinatorics*, chapter Concentration, pages 1–46. Springer, 1998.
30. A. Medina, A. Lakhina, I. Matta, and J. W. Byers. Brite: An approach to universal topology generation. In *9th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pages 346–, 2001.
31. M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2), 2003.
32. M. E. J. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46:323–351, 2005.
33. M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, Jul 2001.
34. I. Norros and H. Reittu. On a conditionally Poissonian graph process. *Advances in Applied Probability*, 38(1):59–75, 2006.
35. D. Peleg and E. Upfal. A trade-off between space and efficiency for routing tables. *J. ACM*, 36(3):510–530, 1989.
36. M. Thorup and U. Zwick. Compact routing schemes. In *SPAA*, pages 1–10, 2001.
37. M. Thorup and U. Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005.
38. B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.