Compact Routing in 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

Abstract. 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 $\varepsilon > 0$. 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.

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 [20]. 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, 28, 29]. 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 [18, 23, 29], planar graphs [19, 25], fixed-minor-free graphs [2], or graphs with low doubling dimension [1, 21, 22]. These topology-specific schemes achieve significant improvements on the stretch-space tradeoff over universal routing schemes.

Power-law graphs [27] 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, 17]. 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. [24] evaluate the universal routing scheme of Thorup and Zwick (*TZ*) [29] 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. [15] 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 [16] 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.

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 [29] 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.

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] 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 [29], 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 [17, 24], our bound is $O(n^{1/12+\varepsilon})$. The routing scheme requires a stretch-5 handshaking (similar to [29, 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. 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 $\tilde{O}(n^{1+\gamma})$ for space and preprocessing of random power-law graphs. Complete proofs of the results in this paper as well as the detailed distance oracle results can be found in a technical report [9].

2 Preliminaries

We adapt the random graph model for fixed expected degree sequence as defined by Aiello, Chung, and Lu [6, 10, 11] 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* **RPLG**(n, τ) *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.

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.

The routing algorithm is inspired by and based on [14, 29]. We also use a set of landmarks $A \subseteq V$, but different from [14, 29], 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 [29], 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 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. $\texttt{header} = (route, s, addr)$*, where* $route = \texttt{toLandmark}$ *and* $addr = \texttt{addr}(t)$*,*
3. $\texttt{header} = (route, s, t, pos, SP)$*, where* $route \in \{\texttt{fromLandmark}, \texttt{direct}\}$*, pos is a non-negative integer that may be modified along the route, and* $SP = SP(s,t)$ *if* $route = \texttt{direct}$ *or* $SP = SP(\ell(t), t)$ *if* $route = \texttt{fromLandmark}$*,*
4. $\texttt{header} = (route, s, t, SP)$*, where* $route = \texttt{handshake}$ *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 $t \in B_G(w) \cup \mathsf{core}_{\gamma'}(G)$.

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 (see [9]). 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*

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

```
if u = s then
    if t ∈ B_G(s) then
        send packet with header = (local, s, t) using port_s(t) stored in tbl(s)
    else if u knows SP(s, t) /* due to handshake */ then
        send packet with header = (direct, s, t, 0, SP(s, t)) using port SP(s, t)[0]
    else
        send packet with header = (toLandmark, s, addr(t)) using port_s(ℓ(t)) stored in
        tbl(s)
    end if
    exit
end if
/* u ≠ s */
if u = header.t then
    exit as the packet arrived.
end if
if header.route = toLandmark then
    if u = header.addr.ℓ(t) then
        header.route ← fromLandmark; header.pos ← 0; header.SP ←
        header.addr.SP(ℓ(t), t);
        forward packet with the new header using port header.SP[0]
    else
        forward the packet to port_u(header.addr.ℓ(t)) stored in tbl(u)
    end if
else if header.route ∈ {fromLandmark, direct} then
    header.pos ← header.pos + 1
    forward the packet using port header.SP[header.pos]
else if header.route = local then
    forward the packet using port_u(header.t) stored in tbl(u)
end if
```

(3) every routing decision takes constant time. In addition, for random graphs sampled from **RPLG**(n, τ)*, 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

Stretch. The proofs use the triangle inequality as in [14, 29].

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 ran-

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

if $\texttt{header}.route = \texttt{fromLandmark}$ **and** $u = \texttt{header}.t$ **and** $\texttt{header}.s \in B_G(u) \cup \mathsf{core}_{\gamma'}(G)$ **then**
  send packet with $\texttt{header} = (\texttt{handshake}, u, \texttt{header}.s, Nil)$ using $\texttt{port}_u(\texttt{header}.s)$ stored in $\texttt{tbl}(u)$.
else if $\texttt{header}.route = \texttt{handshake}$ **then**
  $\texttt{header}.SP = q \cdot \texttt{header}.SP$ /* prepend the port q as part of the reverse path */
  if $\texttt{header}.t = u$ /* reach handshake destination */ **then**
    store $SP(u, \texttt{header}.s) = \texttt{header}.SP$ locally for later use (see Line 4 of LANDMARKBALLROUTING.)
  else
    forward packet with the new `header` to $\texttt{port}_u(\texttt{header}.t)$ stored in $\texttt{tbl}(u)$.
  end if
end if

dom 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 1. *Let G be a random graph sampled from* $\mathbf{RPLG}(n, \tau)$*. The volume $Vol(G)$ satisfies $n < Vol(G) \leq \frac{\tau-1}{\tau-2} n$.*

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. The basic idea to prove the results for the $\mathbf{RPLG}(n, \tau)$ model is to split the random variable for the degree D_i of node v_i into deterministic and random edges and then bound both parts individually.

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

Lemma 3. *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 $Vol(S)/8 \leq vol(S) \leq 4\,Vol(S)$.*

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 4 ([10, Lem. 3.3]). *For any two disjoint subsets S and T with $Vol(S) \cdot Vol(T) > c \cdot Vol(G)$, we have $\Pr[d(S,T) > 1] \leq e^{-c}$.*

Core size. To compute the size of $\mathsf{core}_{\gamma'}(\boldsymbol{w})$, we solve the inequality $w_k > n^{\gamma'}$ and obtain $k < n^{\gamma'(1-\tau)+1}$. 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 2).

Lemma 5. *Let G be a random graph sampled from* **RPLG**(n, τ)*. With probability at least $1 - 1/n^2$ it holds that $\mathsf{core}_{\gamma'}(\boldsymbol{w}) = \{v_i : w_i > n^{\gamma'}\} \subseteq \{v_i : \deg(v_i) > n^{\gamma'}/4\} = \mathsf{core}_{\gamma'}(G)$.*

Lemma 6. *Let G be a random graph sampled from* **RPLG**(n, τ)*. With probability at least $1 - 1/n^2$, $\left|\mathsf{core}_{\gamma'}(G)\right| = \Theta(n^{\gamma})$.*

Ball sizes.

Lemma 7. *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* **RPLG**(n, τ)*, with probability at least $1 - 3/n^2$, it holds that for all $u \in V(G)$,*

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

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

Since for **RPLG**(n, τ) 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 as follows: Let $V' = V \setminus \mathsf{core}_{\gamma'}(\boldsymbol{w})$. Now define $B_0 = \{u\}$ and $B_i = \{v : d_G(u, v) \leq i\}$. 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 $C_i \cup C_{i+1}$.

Lemma 8. *For circle C_i, the following holds with probability at least $1 - 2/n^3$: If $Vol(C_i) < 192 \ln n$, then $E_i \leq 4 \cdot 192 \ln n$, and if $Vol(C_i) \geq 192 \ln n$, then $E_i \leq 4\, Vol(C_i)$.*

Since there are at most n circles, Lemma 8 holds for all circles with probability at least $1 - 2/n^2$.

Table Sizes and Computations. The core $\mathsf{core}_{\gamma'}(G)$ has size $\Theta(n^\gamma)$ with probability at least $1 - 1/n^2$ (Lemma 6) and all balls $B_G(u)$ have size $O(n^\gamma)$ with probability at least $1-3/n^2$ (Lemma 7). Therefore, we have the following result.

Lemma 9. *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

$$TC(G) = O\left(m \cdot \left|\mathsf{core}_{\gamma'}(G)\right| + \sum_{v \in V(G)} T(B_G(u))\right), \tag{2}$$

$$SC(G) = O\left(n \cdot \left|\mathsf{core}_{\gamma'}(G)\right| + \sum_{v \in V(G)} |B_G(u)|\right). \tag{3}$$

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 6); (3) $|B_G(u)| = O(n^\beta)$ for all vertices u (Lemma 7); (4) $T(B_G(u)) = O(n^\beta \log n)$ for all vertices u (Lemma 7). Therefore, from Equations (2) and (3), 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 $1 - 1/n$. □

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

By Lemma 10, 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 $O(\log n \log \log n)$, as proven in the following lemma.

Lemma 11. *For a random graph G sampled from* **RPLG**(n, τ)*, 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* **RPLG**(n, τ) *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.

Lemma 12 (Chung and Lu [10, Claim 4.1]). *Let G be a random graph sampled from* **RPLG**(n, τ)*. The diameter of the subgraph induced by* $\mathsf{core}^+(\boldsymbol{w})$ *in G is* $O(\log \log n)$ *with probability at least* $1 - 1/n$*.*

Lemma 13 (Chung and Lu [10, Claim 4.2]). *Let G be a random graph sampled from* **RPLG**(n, τ)*. 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 13). *Let G be a random graph sampled from* **RPLG**(n, τ)*. Let C be the constant in Lemma 13. 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 11). 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 13. 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 2, 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 10), the rest of the shortest path can also be encoded with $O(\log n \log \log n)$ bits, with probability $1 - o(1)$. $\square$

5 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.

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 $\tau \in (2, 3)$). In addition, we generated graphs of 10,000 nodes with the tool BRITE [26] using the configurations for the Barabási [7] and Waxman [30] models for an Autonomous System Topology (AS) and a Router Topology (RT). 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 **RPLG**(n, τ).

Routing schemes. In the specification of our routing scheme LANDMARK-BALLROUTING, 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 [26] 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. We compare our high-degree selection strategy with the random selection with probability $n^{-1/2}$, which is *similar* to Thorup and Zwick [29] 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,

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

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 [29]. 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.

6 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 random power-law graphs models is an interesting topic.

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. W. Chen, C. Sommer, S.-H. Teng, and Y. Wang. A compact routing scheme and approximate distance oracle for power-law graphs. Technical Report MSR-TR-2009-84, Microsoft Research, July 2009.
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. M. Enachescu, M. Wang, and A. Goel. Reducing maximum stretch in compact routing. In *INFOCOM*, pages 336–340, 2008.
16. P. Erdős and A. Rényi. On the evolution of random graphs. *Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei*, 5:17–61, 1960.
17. 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.
18. P. Fraigniaud and C. Gavoille. Routing in trees. In *ICALP*, pages 757–772, 2001.
19. C. Gavoille and N. Hanusse. Compact routing tables for graphs of bounded genus. In *ICALP*, pages 351–360, 1999.
20. C. Gavoille and S. Perennes. Memory requirements for routing in distributed networks (extended abstract). In *PODC*, pages 125–133, 1996.
21. G. Konjevod, A. W. Richa, and D. Xia. Optimal-stretch name-independent compact routing in doubling metrics. In *PODC*, pages 198–207, 2006.
22. G. Konjevod, A. W. Richa, D. Xia, and H. Yu. Compact routing with slack in low doubling dimension. In *PODC*, pages 71–80, 2007.
23. A. Korman. Improved compact routing schemes for dynamic trees. In *PODC*, pages 185–194, 2008.
24. D. V. Krioukov, K. R. Fall, and X. Yang. Compact routing on internet-like graphs. In *INFOCOM*, 2004.
25. 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.
26. 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.
27. M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2), 2003.
28. D. Peleg and E. Upfal. A trade-off between space and efficiency for routing tables. *J. ACM*, 36(3):510–530, 1989.
29. M. Thorup and U. Zwick. Compact routing schemes. In *SPAA*, pages 1–10, 2001.
30. B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.