Compact Routing on Power Law Graphs with Additive Stretch*

Arthur Brady† Lenore Cowen†

Abstract

We present a universal routing scheme for unweighted, undirected networks that always routes a packet along a path whose length is at most an additive factor of d more than OPT (where OPT is the length of an optimal path), using $O(e \log^2 n)$-bit local routing tables and packet addresses, with d and e parameters of the network topology. For power-law random graphs, we demonstrate experimentally that d and e take on small values. The Thorup-Zwick universal multiplicative stretch 3 scheme has recently been suggested for routing on the Internet inter-AS graph; we argue, based on the results in this paper, that it is possible to improve worst-case performance on this graph by directly exploiting its power-law topology.

1 Introduction

Compact routing refers to the design of routing algorithms which store a small amount of information in a routing table at each node in a network, and provide a bound on the *stretch* of messaging routes. Following the terminology of Elkin for graph spanners [12], we say that a compact routing scheme has *stretch* (α, β) when the length of the route taken by a message from a source node u to a destination v is always at most $\alpha d(u, v) + \beta$, where $d(u, v)$ is the minimum length of a path from u to v in the network. A scheme with stretch $(\alpha, 0)$ is said to have *multiplicative stretch* α; one with stretch $(1, \beta)$ is said to have *additive stretch* β. If the worst-case size of a local routing table in a given scheme is $o(n)$, the scheme is said to be *compact*, and there is typically a tradeoff between minimizing multiplicative stretch and minimizing routing table size. Compact routing (and the closely-related problems of spanner construction and distance labeling) has been well-studied on special-case networks, such as trees [16, 27], graphs of bounded genus [19], and, recently, graphs whose doubling dimension [7] is bounded. There has also been much recent work on *universal* compact routing schemes [1, 10, 15, 27], which provide space bounds and multiplicative stretch guarantees for any undirected network. (As far as we know, the only work on compact routing in directed networks is work on roundtrip routing, appearing in [9, 26]; though additive stretch has been studied for graph spanners [3, 6], we are unaware of any prior compact routing results, even for special-case networks.)

A great deal of recent research has focused on discovering and modeling the topological properties of various large-scale real-world networks, including social networks and the Internet graph. In 1999, several teams [5, 20] reported independently that the degree distribution of the web graph appeared to follow a power-law. In a seminal paper of the same year, Faloutsos et al. [13] reported that the degrees of the Internet inter-AS graph also appeared to follow a power law. Since then, there has been extensive interest in random-graph models which capture this property. Two popular families of models have been proposed by Barabási and Albert [5] and Aiello, Chung and Lu [2]. Barabási and Albert proposed a model in which new vertices are added iteratively to a graph and linked to existing vertices with probability proportional to their degree; they showed that this construction induces a power-law degree distribution in the resulting graph. Aiello et al. proposed a power-law random graph model in which the degree sequence of an n-node graph is constructed according to the desired distribution, and demonstrate an elegant method of building a graph using the given degree sequence. In what follows, we will refer to graphs generated using the Barabási and Albert model as PC random graphs, and we will call graphs generated by the Aiello, Chung and Lu model PLRG graphs.

The minimum multiplicative stretch achieved by any known universal compact routing scheme is 3 [10, 27]. In the *name-independent* routing model (cf. [1, 4]), as well as in the case of *name-dependent* models where packet headers are restricted to be precisely $\log n$ bits long, a result of Gavoille and Gengler [18] shows that this is also a lower bound. Universal routing schemes only make their guarantees based on worst-case graph topologies; the question naturally arises as to the stretch *in practice* of these schemes on models of real-world networks, in particular on power-law networks. This question motivated the recent work of Krioukov et al. [22], who showed that on power-law random

*Supported in part by NSF grant CCR-0208629.

†Department of Computer Science, Tufts University, Medford, MA 02155. Email: {abrady,cowen}@cs.tufts.edu

graphs generated by the PLRG model, the average experimental performance of Thorup and Zwick's universal multiplicative stretch 3 routing algorithm [27] is in fact much better than three, and closer to 1.1 on certain power-law graphs. We consider the same problem they do (name-dependent compact routing on PLRG models) but show that designing schemes that exploit the expected structure of these networks can lead to better stretch on this class of networks.

While the observation that the degrees of the actual Internet inter-AS routing graph obey a power-law distribution is grounded in a lot of research (both theoretical and based on empirical measurement) [14, 24], work has also appeared recently that has convincingly cast doubt on the full sufficiency of this type of Internet model. Chen et al. [8] observe that the inferred power-law distribution may be (at least in part) due to artifacts of measurement, and Willinger et al. [29] conclude that descriptive models in their current form fail to address the underlying causes of the observed emergent properties of the network. Despite this, the idea of a power-law network remains an object of broad interest, making appearances in economic, social and biological models as well as in computer science.

The main theoretical result of this paper is a universal routing scheme on undirected, unweighted graphs with *additive* stretch $(1, d)$, using $O(e \log^2 n)$-bit local routing tables and message headers, where d and e are parameters of the network. We also describe a hybrid compact routing scheme which overlays our stretch $(1, d)$ scheme with the Thorup-Zwick universal stretch $(3, 0)$ scheme, with table size the same as in the latter (namely $O(\sqrt{n} \log^2 n)$). The hybrid scheme always routes along the best path provided by either scheme; its stretch is thus $\min\{(1, d), (3, 0)\}$.

Using the PLRG model, we verify experimentally that for power-law random graphs generated over a significant range of power-law parameters (which includes all values which have been estimated for the Internet graph), d and e take on small constant values, suggesting that our new routing scheme may work very well in practice.

Part of our contribution is the development of DIGG (DynamIc Graph Generator), a free C++-based software suite for the efficient generation and representation (in XML) of large graphs according to user-supplied parameters and generation algorithms. As the name suggests, the software also provides a way to create, store and analyze the life-cycles of *dynamic* graphs; this functionality was not used in the current paper, but we intend to demonstrate its utility in future work. The current (beta) version of the DIGG source code, the library of graphs which we generated for the experimental component of this paper, and the analytic code we created for our experiments are all freely available at **http://digg.cs.tufts.edu**.

2 Definitions

Consider a communications network modeled as a connected, undirected, unweighted graph $G = (V, E)$, $|V| = n$, with network nodes represented as vertices (each of which is assigned a unique label $v \in \{1, 2, \ldots, n\}$), and direct communications links represented as edges $uv \in E$.

A *routing scheme* R is a distributed algorithm defined on G which guarantees that any vertex u can send a message M to any specified destination v (along some (u, v)-path P in G), using metadata stored in M along with information stored locally at each vertex in P.

We refer to the metadata stored (by R) in a message M as M's *header*, and to the local information stored at a vertex v as v's *routing table*. Given an input graph $G = (V, E)$, a routing scheme R must specify:

1. the construction of the routing table at each vertex $v \in V$,
2. the construction of the header of any message M originating at a given source u and intended for a given destination v, and
3. a forwarding function $F(\mathit{table}(x),\ \mathit{header}(M))$ computed locally at each vertex $x \in V$ which, given the information in x's routing table and the information in M's header, selects an edge adjacent to x along which to forward M.

F is known as R's *routing function*. Given a source vertex u, a destination vertex v and a message M, the sequence of vertices $\langle u = v_0,\ v_1,\ \ldots,\ v_k \rangle$ defined by successive applications of $F(\mathit{table}(v_i),\ \mathit{header}(M))$ must be such that k is finite, and $v_k = v$. We refer to this (u, v)-path as the *route* P_{uv} from u to v with respect to R. Note the distinction between the *label* of a vertex v of G and the *header* of a message destined for v. Because the header of a message is forced to be succinct but is otherwise unconstrained, we are studying compact routing in the *name-dependent* routing model (also known as the *labeled* routing model). In contrast, routing models in which a message header destined for v is constrained to contain only the $(\log n)$-bit label of v are known as *name-independent* models (cf. [1, 4] for formal definitions and discussion). Our compact routing scheme requires neither a port-relabeling nor rewritable headers, so it is said to be a *1-phase* [11] routing scheme in the *fixed-port* [16] model.

As in Elkin [12], we say that a routing scheme R has *stretch* (α, β) with respect to a family of graphs G_f if the length of any route P_{uv} is always at most $\alpha d(u, v) + \beta$, where $d(u, v)$ is the minimum length of a path from u to v in the network.

A routing scheme is called *compact* if headers have size bounded by $O(\log^c n)$ bits for some constant c, and local routing tables each have size bounded by $o(n)$ bits. There is a natural tradeoff between header size and routing table size. There is also a natural tradeoff between the total size of all data structures employed by a routing scheme and its stretch; intuitively, if we are willing to take longer paths, we can use less information overall to get where we're going.

Recent experimental work of Krioukov et al. [22] looked at the performance of the best known universal multiplicative stretch-3 compact routing algorithm (due to Thorup and Zwick) on graphs generated by the PLRG model, and showed that the *average* stretch of routes on these graphs was much better than 3, and was closer to a multiplicative stretch of 1.1. The question we asked that led to this paper was: can you do better still on PLRGs?

2.1 Some facts about power-law graphs

Definition 2.1. *A* power-law graph $G = (V, E)$ *is an undirected, unweighted graph whose degree distribution approximates a power law, i.e. the number* $y = |\{v \in V \mid deg(v) = x\}|$ *of vertices whose degree is* x *satisfies*

1. $\begin{cases} y = \lfloor c \rfloor - r & \textit{when } x = 1 \\ y = \lfloor \frac{c}{x^\gamma} \rfloor & \textit{when } x = 2,\ 3,\ \ldots,\ \left\lfloor c^{\frac{1}{\gamma}} \right\rfloor \end{cases}$,
2. $r = n - \sum_{x=1}^{\left\lfloor c^{\frac{1}{\gamma}} \right\rfloor} \lfloor \frac{c}{x^\gamma} \rfloor$, *and*
3. c *is a value minimizing* $|n - r|$

for some constant $\gamma \in \mathcal{R}^+$*, called the* power-law parameter *of* G*.*

Definition 2.2. *A* γ-RPLG *(for "random power-law graph") is a graph* $G_\gamma = (V, E), |V| = n$*, which has been uniformly randomly selected from the set of all* n*-vertex power-law graphs with power-law parameter* γ*.*

Using a model very close to the PLRG model (but not exactly identical, see [2, 23] for discussion); Lincoln Lu [23], in his probabilistic analysis of power-law graph topology, observed that for sufficiently large values of n, with high probability[1] the following hold:

[1] When we say "X is true with high probability," we mean that the probability that X is false is $o(n^{-1})$.

1. For all ranges of $\gamma > 0$, a random power-law graph G_γ has a unique giant component, and all components other than the giant component have size at most $O(\log n)$.

   (Throughout the following – since we consider a routing problem on connected networks, and since all non-giant components are small – we ignore all non-giant components, and abuse notation slightly by identifying G_γ with its giant component.)

2. For $0 < \gamma < 2$, G_γ contains an edge-dense "core" of diameter at most 3, which is connected to a set of "tree-like tails" of constant length.

3. For $2 < \gamma < 4$, G_γ contains
   - an inner layer consisting of a small edge-dense "core,"
   - an outer layer of "tree-like tails,"
   - and a "middle layer" in between the core and the outer layer.

   Furthermore, each of these three layers is of diameter $\Theta(\log n)$.

4. For all ranges of γ, the highest-degree vertices in G_γ are contained in the "core."

These facts suggest considering routing algorithms that employ different strategies in the core and in the tails, which is what we do in the next section.

3 Our new routing scheme

Definition 3.1. *Let* G *be an undirected, unweighted graph, and let* h *be the node of* G *of highest degree (breaking ties lexographically by node names, so that* h *is always uniquely defined). For each positive even integer* d*, we define the* d-core *of* G *to be the subgraph of* G *induced by the set of vertices of distance at most* $d/2$ *from* h *(thus the* d*-core is of diameter* d*). We also define the* d-fringe *of* G *to be the subgraph of* G *induced by the set of vertices which are not in the* d*-core of* G*.*

Definition 3.2. *Given a graph* H*, we define the* extra-edge count e_H *of* H *to be the minimum number of edges that must be removed from* H *in order to make* H *acyclic.*

With the parameters d and e defined, we are now ready to state our principal theorem:

Theorem 3.1. *For any (unweighted, undirected) graph* $G = (V, E), |V| = n$*, there is a routing scheme* CFROUTE *that uses* $O(e \log^2 n)$*-bit headers and routing tables, and has a stretch of* $(1, d)$*, where* e *is the extra-edge count of the* d*-fringe of* G*.*

Notice that it is easy to construct graph families for which either d or e must always be large, meaning that there exist graph families where this routing scheme isn't even compact. In this section, we prove this theorem. In the following section, we describe an experimental study of the tradeoffs between d and e for the PLRG model, and demonstrate ranges of γ for which we will expect to achieve small *additive* stretch using this routing scheme.

In order to present CFROUTE, we first need to slightly modify a known compact routing algorithm on trees, which we do in subsection 3.1. In subsection 3.2, we present and analyze CFROUTE as a proof of Theorem 3.1.

3.1 A compact routing algorithm for trees

There exist compact routing schemes for trees due both to Fraignaud and Gavoille [16] and to Thorup and Zwick [28] which use $O(\log^2 n)$-bit headers, $O(\log n)$-bit routing tables, and guarantee stretch $(1, 0)$ (that is, they always route along optimal paths). Given a tree $T = (V, E)$, we denote by $TZ_T table[u]$ the local routing table assigned to vertex u by the Thorup-Zwick tree-routing scheme on T, and by $TZ_T header[w]$ the header assigned to vertex w by the Thorup-Zwick tree-routing scheme on T. Let $d_T(u, v)$ represent the length of the unique (u, v)-path in T.

Peleg [25] demonstrated that given any tree T with uniform edge weights, each vertex v of T can be assigned an $O(\log^2 n)$-bit label $l(v)$, such that given the labels of any two vertices v, w in T, the distance $d(v, w)$ between them can be computed exactly. Such a labeling scheme is referred to as an *(exact) distance labeling scheme;* we will refer to the scheme in [25] as the Peleg scheme. We denote by $l_T(v)$ the label generated by the Peleg scheme for a vertex v in a tree T.

We augment the Thorup-Zwick tree-routing scheme slightly to create a new tree-routing scheme TZ' as follows.

Let $T = (V, E)$ be any tree with root r.

1. Initially, store a routing table $TZ_T table[u]$ at each node u, and let $TZ_T header[w]$ be the header of node w, exactly as in the Thorup-Zwick tree-routing scheme.
2. For each vertex u, add the label $l_T(u)$ to $TZ_T table[u]$ to form $TZ'_T table[u]$.
3. For each vertex v, add the label $l_T(v)$ to $TZ_T header[v]$ to form $TZ'_T header[v]$.
4. Routing decisions are made exactly as prescribed by the Thorup-Zwick tree-routing scheme.

LEMMA 3.1. *TZ' uses $O(\log^2 n)$-bit headers, $O(\log^2 n)$-bit routing tables, and allows any vertex u, given the header of any destination v, to compute $d_T(u, v)$.*

Proof. The header and routing table size bounds are immediate from the fact that TZ' only requires $O(\log^2 n)$ more bits than the corresponding headers and tables in the original scheme. $d_T(u, v)$ can be computed from $l_T(u)$ and $l_T(v)$ exactly as described in [25]. $\square$

3.2 Proof of Theorem 3.1

Given an unweighted, undirected graph $G = (V, E)$, $|V| = n$, and an even integer d, denote the d-core of G by I, and denote the d-fringe of G by F. Let h be the vertex of G of highest degree, with ties broken lexicographically. Let T be a single-source shortest path tree spanning G, with source h. Consider $T \cap F$. Extend $T \cap F$ by adding edges between vertices in F, until we have a spanning tree on each connected component of F. Call the resulting forest T_F. Let $E' = \{uv \mid uv \in E,\ uv \notin T_F,\ u, v \in F\}$ be set of all edges of G between vertices in F which are not contained in T_F. Let e_F denote the extra-edge count of F; note that because T_F spans each connected component of F, $|E'| \leq e_F$.

For any $u, v \in V$, let $d(u, v)$ denote the distance from u to v in G, let $d_T(u, v)$ represent the distance from u to v in T, and let $d_{T_F}(u, v)$ represent the distance from u to v in T_F, or ∞ if there is no (u, v)-path in T_F.

DEFINITION 3.3. *We define a set $\mathcal{T}$ of trees T_i to be the union of the following two sets:*

1. *a set of spanning trees $\{T_0 = T,\ T_1,\ \ldots,\ T_{|E'|}\}$ on G, constructed as follows:*
   - *$i \longleftarrow 1$.*
   - *For each edge $uv \in E'$,*
     - *Grow a single-source shortest path tree T_i rooted at u which includes uv.*
     - *Increment i.*
2. *the connected components of T_F, with each assigned an arbitrary root vertex.*

Note that each vertex of G is in at most one component of T_F. Throughout the following we refer to an element of $\mathcal{T}$ as T_i.

CFROUTE consists of four parts: a preprocessing step, in which we construct temporary data structures using TZ', a labeling step, in which nodes are assigned headers, a storage step, in which a local routing table is constructed at each node, and the routing procedure itself.

$$\begin{array}{l} \text{header}[v] \longleftarrow \emptyset \\ \text{For each } T_i \in \mathcal{T} \mid v \in T_i,\ 0 \leq i < |\mathcal{T}| \\ \quad \text{header}[v] \longleftarrow \text{header}[v] \circ (i,\ TZ'_{T_i} header[v]) \end{array}$$

Figure 1: **The labeling step**

$$\begin{array}{l} \text{table}[u] \longleftarrow \emptyset \\ \text{For each } T_i \in \mathcal{T} \mid u \in T_i,\ 0 \leq i < |\mathcal{T}| \\ \quad \text{table}[u] \longleftarrow \text{table}[u] \circ (i,\ TZ'_{T_i} table[u]) \end{array}$$

Figure 2: **The storage step**

3.2.1 Preprocessing

Process each tree $T_i \in \mathcal{T}$ using TZ'. Let $TZ'_{T_i} table[u]$) be the routing table for node u in tree T_i, and let $TZ'_{T_i} header[v]$) be the header assigned to node u in tree T_i.

3.2.2 Labeling

Assign to each vertex v a list of pairs $(i,\ TZ'_{T_i} header[v])$ (ordered by increasing i), one for every tree T_i which contains v.

3.2.3 Storage

The routing table stored at each vertex u consists of a list of pairs $(i,\ TZ'_{T_i} table[u])$ (ordered by increasing i), one for every tree T_i which contains u.

LEMMA 3.2. CFROUTE *uses $O(e_F \log^2 n)$-bit headers and routing tables.*

Proof. Given any node $v \in G$, the number of trees $T_i \in \mathcal{T}$ containing v is at most $e_F + 2$:

- v is contained in T (because T spans G),
- v is contained in $|E'| \leq e_F$ spanning trees T_i in $\mathcal{T}$ by construction, and
- v is contained in at most 1 component of T_F.

So since the routing table at each node u contains at most $e_F + 2$ entries $(i,\ TZ'_{T_i} table[u])$, and since the header assigned to each node v contains at most $e_F + 2$ entries $(i,\ TZ'_{T_i} header[v])$, the result follows immediately from Lemma 3.1. □

3.2.4 Routing procedure

Routing from a source vertex u to a destination v using CFROUTE proceeds as follows:

1. For each tree T_i containing both u and v, extract $l_{T_i}(u)$ from $TZ'_{T_i} table[u]$, extract $l_{T_i}(v)$ from $TZ'_{T_i} header[v]$, and compute $d_{T_i}(u, v)$ according to the Peleg scheme ([25]).
2. Choose some tree T_j such that $d_{T_j}(u, v)$ is minimized.
3. Route from u to v in T_j according to TZ'.

3.2.5 Analysis of the routing procedure

LEMMA 3.3. *Given any node u and any destination v,* CFROUTE *routes from u to v along a path of length at most $d(u, v) + d$.*

Proof. Let u and v be any two vertices which are both in I. Since $d_T(u, v) \leq d_T(u, h) + d_T(h, v) = d(u, h) + d(h, v) \leq \frac{d}{2} + \frac{d}{2} = d$, and since $1 \leq d(u, v)$, we have that $d_T(u, v) \leq d(u, v) + (d - 1) < d(u, v) + d$.

Now let u and v be such that $u \in I$ and $v \notin I$. Since $u \in I$, $d(u, h) \leq \frac{d}{2}$. Since $d(h, v) \leq d(h, u) + d(u, v) \leq \frac{d}{2} + d(u, v)$, we have that $d_T(u, v) \leq d_T(u, h) + d_T(h, v) = d(u, h) + d(h, v) \leq \frac{d}{2} + [\frac{d}{2} + d(u, v)] = d(u, v) + d$.

Finally, let u and v be any two vertices both in F, and let P be any shortest (u, v)-path in G.

Either $d_{T_F}(u, v) = d(u, v)$, or $d_{T_F}(u, v) > d(u, v)$. If the latter is the case, then there exists some edge $u'v' \in P$ which is not in T_F. We consider two cases.

1. If there exists some such edge where $u', v' \in F$, then because $u'v' \notin T_F$, $u'v' \in E'$, so the preprocessing routine of CFROUTE constructed a single-source shortest path tree T_i spanning G with source u' (or v'; wlog assume u'). Since we have $d_{T_i}(u, v) \leq d_{T_i}(u, u') + d_{T_i}(u', v) = d(u, u') + d(u', v)$ and since u' is on some shortest (u, v)-path P in G, we conclude that $d_{T_i}(u, v) = d(u, v)$ for some T_i.
2. Now assume that $u' \in I$ or $v' \in I$ (or both) for all edges $u'v' \in P$ which are not in T_F.

   Notice that because P contains at least one vertex in I, we have that $d(u, I) + d(I, v) \leq |P| = d(u, v)$. So we conclude that $d_T(u, v) \leq d_T(u, h) + d_T(h, v) \leq [d(u, I) + \frac{d}{2}] + [\frac{d}{2} + d(I, v)] \leq d(u, v) + d$.

TZ' routes with stretch $(1, 0)$ on each tree T_i. We have shown that for any two vertices $u, v \in V$, there is always some tree $T_i \in \mathcal{T}$ such that $d_{T_i}(u, v) \leq d(u, v) + d$. Given a source u and a destination v, since we always choose to route within a tree T_i minimizing $d_{T_i}(u, v)$, we have that any (u, v)-route in CFROUTE has length at most $d(u, v) + d$, giving a stretch of $(1, d)$ as desired. □

3.3 A hybrid scheme In practice, when faced with a network that may or may not be a PLRG, we remark that the right thing to do is to superimpose our scheme and Thorup and Zwick (TZ)'s *universal* stretch $(3, 0)$ routing scheme, resulting in a hybrid scheme. (Note that the latter is a different scheme from the Thorup-Zwick tree-routing scheme discussed in Section 3.1). The key observation is that the universal TZ scheme can be modified so that a packet that arrives at any node u destined for some node v can, with help from the local routing table stored at u, compute exactly the length of the path from u to v that would be traversed using that scheme. (Our scheme already uses an analogous function.) The details are straightforward and are omitted from this extended abstract.

Thus we can simply concatenate headers for both schemes, and store tables for both schemes at every node. A packet then decides which scheme to follow by computing which one would result in a shorter path to its destination, then routing according to that scheme. The hybrid scheme is guaranteed to have better stretch than either the universal TZ scheme or our new scheme implemented separately: its stretch is $\min\{(1, d), (3, 0)\}$. It also uses tables of the same asymptotic size as those used by the universal TZ scheme, namely $O(\sqrt{n} \log^2 n)$ bits.

4 Experiments

We have shown that for a given graph G, a given value of d, and I and F being the d-core and d-fringe of G respectively, CFROUTE guarantees an additive stretch of $(1, d)$ and uses $O(e_F \log^2 n)$-bit headers and routing tables. It remains to determine what values of d and e_F are typical for power-law graphs.

We conducted two sets of experiments: the first was an exploration of RPLG topology, and the second compared the performance of the Thorup-Zwick universal scheme, our scheme, and the hybrid scheme outlined in Section 3.3 on RPLGs.

Our topology experiments were designed to provide answers to the following:

- For a given γ-RPLG G_γ, can we find a small value of d such that the extra-edge count e_F of the d-fringe F of G_γ is also small?
- Lu [23]'s results on the properties of power-law graphs hold for sufficiently large n. We were interested in whether or not the desirable properties (such as a low-diameter core) were observed for the graph sizes we were interested in.

For the topology experiments, we used DIGG to implement the PLRG generator in [2], with which we generated random power-law graph instances on n vertices (for $n = 2500$, 5000, 10000, 20000 and 40000) for power-law parameter γ (for $1.2 \leq \gamma \leq 3.0$). We constructed 30 graph instances for each value of the parameter pair (n, γ). DIGG supports the fast saving and loading of graph instances to and from XML files; we were thus able both to create a permanent library of all the graph instances we generated, and to provide a basis for exact or parallel replication of our experiments by others.

For each generated graph instance G_γ, we calculated[2] the following:

1. d_{min}, the minimum value of d such that the d-fringe of G_γ was exactly a forest, and
2. for each $1 \leq d \leq d_{min}$,
    - the extra-edge count e of the d-fringe of G_γ.

We divided our topology experiments into two phases. In phase 1, we looked at graphs G_γ where $1.2 \leq \gamma \leq 1.9$; in phase 2, we examined G_γ for $2.0 \leq \gamma \leq 3.0$. According to the predictions in [23], we expected to find that graphs in the first range would actually have cores of *constant* diameter, and that graphs in the second range would have core of diameter $O(\log n)$. The extremely slow growth rates of the observed diameters for both types of graph this seem to confirm this prediction.

The table size of our algorithm is acceptable when e is not too large, and its stretch is smallest when $d = 2r$ is as small as possible. Define $\overline{e}$ to be the average value of the extra-edge count e of the d-fringe of G_γ (for given values of n, γ and d) across all graphs G_γ in the corresponding sample set; we denote by σ_e the estimated standard deviation of this statistic.

In phase 1 ($1.2 \leq \gamma \leq 1.9$), we found that for *all* observed values of n and γ, there was a sharp threshold: $\overline{e}$ was large when d was set to be 2 or 4, but setting $d = 6$ produced values of $\overline{e}$ and σ_e which were both less than 1. (In other words, for all values studied, the 6-fringe of G_γ differed on average from a forest by less than one edge.)

In phase 2 ($2.0 \leq \gamma \leq 3.0$), we found that for all observed values of n and for $2.0 \leq \gamma \leq 2.5$ (this being the range containing the vast majority of power-law parameter estimates for the various models of the

[2] Exceptions: we generated graphs for ($n = 20000$, $\gamma = 1.2$), but could not analyze them due to memory constraints. Also because of memory issues, for $n = 40000$, we only generated graphs for $2.0 \leq \gamma \leq 3.0$. As our data will show, however, the properties under consideration began to conform very closely to their predicted values in [23] well before n became this large.

Internet and web graphs), setting $d = 10$ produced values of $\overline{e}$ and σ_e which were both less than 5. (In other words, for these values, the 10-fringe of G_γ differed on average from a forest by less than 5 edges.)

Thus we conclude that for graph sizes up to 40,000, our scheme displays a worst-case additive stretch of $(1, 6)$ for phase 1 graphs, and $(1, 10)$ for phase 2 graphs, while maintaining $O(\log^2 n)$-bit tables. Compare to the Thorup-Zwick multiplicative-stretch scheme which uses $O(\sqrt{n}\log^2 n)$-bit tables and has worst-case stretch $(3, 0)$. Note that the theory of [23] implies that for phase 1 graphs, the additive stretch and table size of our scheme is unlikely to increase much as n grows beyond 40,000, whereas for phase 2 graphs, additive stretch and table size should increase logarithmically.

As was noted by Krioukov et al. in [22], the average stretch of the Thorup-Zwick scheme is considerably better than worst-case stretch on power-law graphs. In our routing experiments, simulations of the Thorup-Zwick scheme on our synthetic RLPGs yielded average-case (multiplicative) stretch between 1.25 and 1.18 (for γ between 2.0 and 2.2).[3] Simulations of the additive scheme presented in this paper produced an average multiplicative stretch between 1.22 and 1.11 (for γ in the same range), and the hybrid scheme outlined in Section 3.3 resulted in an average multiplicative stretch between 1.13 and 1.07. See Figures 9 - 11 for details.

The observed average stretch of our scheme was consistently better than the average stretch of the TZ scheme; also, the margin of improvement increased significantly as γ increased, which is intuitively due to the fact that the fringe becomes more sparse as the exponent of the power-law increases.

The hybrid scheme significantly outperformed both schemes (indicating that the sets of optimal routes discovered by each scheme were different from one another, so that when taken together, they provided a strong improvement over either scheme on its own).

We present two sets of figures summarizing two different perspectives on our topology results.

In Figures 3 and 4, n is fixed at a particular value in {10000, 20000}. Within each chart, $\overline{e}$ is plotted as a function of γ and d.

In each of figures 5 - 8, γ is fixed at a particular value in {2.0, 2.1, 2.2, 2.3}. (We emphasize the middle range here because it is in this range that power-law parameters for the Internet inter-AS topology have been estimated.) Within each chart, $\overline{e}$ is plotted as a function of n and d.

For our routing experiments, we simulated the Thorup-Zwick scheme, our scheme, and the hybrid scheme outlined in Section 3.3 on 15 of the 10,000-node graphs which we generated for the topology experiments: 5 graphs each of $\gamma \in \{2.0, 2.1, 2.2\}$.

The essential power of our algorithm lies in exploiting the sparsity of each RPLG, outside the core, using a small number of spanning trees. We discovered in our routing simulations that while a single spanning tree sourced at the highest-degree node provided an average stretch close to the worst-case bound, adding a very small number of spanning trees sourced at fringe edges caused the stretch to drop dramatically. We therefore added a heuristic to the simulations of our scheme and the hybrid scheme: if the extra-edge count of the fringe was less than 5, we added up to 5 trees spanning G_γ, sourced at random edges in the fringe.

Figures 9 - 11 describe our results: we measured the mean stretch over all possible paths (framed both multiplicatively and additively), as well as the percentage of optimal paths used by each algorithm. All data has been averaged over the 5 graphs in each set. More detailed data, including the full distribution of stretch over all paths for each algorithm on each graph, is available on our website.

The most commonly studied model of Internet routing is the *inter-AS graph*, which represents ASs (autonomous systems, which are roughly equivalent to ISPs) as vertices, and communications links (BGP peering links; cf. [17] for discussion) as edges. Recent empirical studies of the inter-AS graph [24] suggest that it has less than 20,000 nodes; its power-law parameter γ has been estimated to be between 2.0 and 2.5. Our choices of n and γ are therefore realistic for this type of model, and our algorithm remains quite compact (table size $O(\log^2 n)$) across observed values of n and γ, guaranteeing an additive stretch of $(1, 10)$.

We remark that a more granular representation of the Internet as a graph would model individual *routers* as vertices. There are approximately 250,000 Internet routers at present [21]. A model of this size would be more than 6 times larger than the graphs we studied in our experiments, but based on extrapolations of the observed growth curves of $\overline{e}$ for phase 2 graphs with γ between 2.0 and 2.5, we predict that when $d = 10$, both $\overline{e}$ and σ_e will remain less than 5 for graphs of this size. Thus for RPLGs of this size, the stretch and table size of our algorithm would remain $(1, 10)$ and $O(\log^2 n)$, respectively. Unfortunately, less is known about the topology of this type of Internet model; it may not even exhibit a power-law degree distribution.

While additive-stretch and multiplicative-stretch

[3] Note that [22] reports an average TZ stretch of 1.1 on RPLGs in this range, whereas we observed stretch closer to 1.2. The disparity is attributable to slight differences in the respective methods used to generate degree distributions for synthetic RPLGs.

schemes are hard to rank against each other, certainly we achieve a dramatic reduction in table size. If we have space resources which can accomodate $O(\sqrt{n}\log^2 n)$-bit tables, we can instead implement the hybrid scheme of Section 3.3, which is always guaranteed to do as well in stretch as the better of our additive and Thorup and Zwick's multiplicative stretch schemes. According to our simulations, the hybrid scheme appears to significantly outperform both schemes in practice.

We have made copies of our code for PLRG generation, topology analysis, routing simulations, and routing-scheme stretch analysis, as well as complete tabulations of all raw and aggregate analysis, and all generated graphs (encoded in XML) available at **http://digg.cs.tufts.edu**.

5 Discussion and future work

Taking into account the unique topological properties of power-law graphs has allowed us to design better compact routing schemes with superior performance on these graphs. We intend to investigate applications of this approach to other graph topologies.

Xenofontas Dimitropoulos and Dmitri Krioukov at CAIDA (**http://www.caida.org**) have recently provided us with graph data which incorporates estimates of inferred AS relationships in the real Internet inter-AS graph, and we intend to study new schemes which take advantage of their particular topological properties.

6 Acknowledgements

Thanks to Daniel Wolchonok and Patrick Schmid for major contributions to the development of DIGG. Kofi Laing and Alva Couch contributed helpful preliminary discussions concerning its high-level design. Dmitri Krioukov and k claffy of CAIDA (**http://www.caida.org**) continue to give critical insight about routing on "real" Internet graphs. Thanks to the anonymous referees for pointing out two errors in an early version of this manuscript. The authors gratefully acknowledge support from NSF grant CCR-0208629.

References

[1] Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, Noam Nisan, and Mikkel Thorup. Compact name-independent routing with minimum stretch. In *Proceedings of the 16th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2004)*, pages 20–24. ACM Press, June 2004.

[2] William Aiello, Fan Chung, and Linyuan Lu. A random graph model for massive graphs. In *STOC '00: Proceedings of the thirty-second annual ACM*

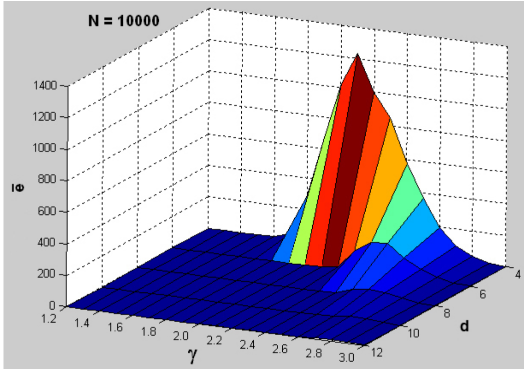

Figure 3: $N = 10000$. In this and the next figure, the mean extra-edge count $\bar{e}$ (over all graphs in each set) of the d-fringe of each graph is plotted as a function of the core diameter d and the power-law parameter γ.

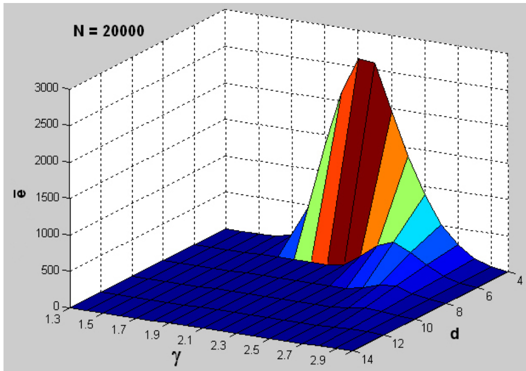

Figure 4: $N = 20000$.

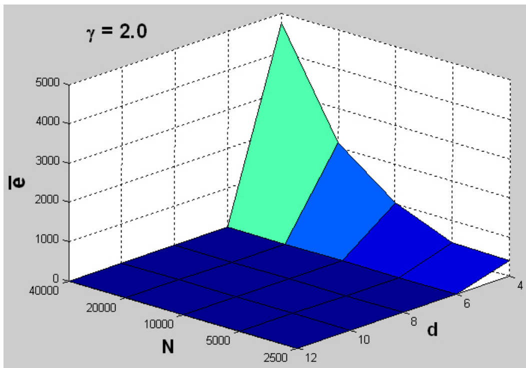

Figure 5: $\gamma = 2.0$. In this and the following three figures, the power-law parameter γ is held fixed, and the mean extra-edge count $\bar{e}$ of each set of graphs is plotted as a function of the graph size N and the core diameter d.

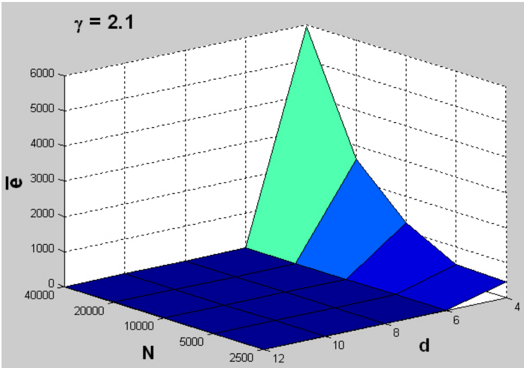

Figure 6: $\gamma = 2.1$.

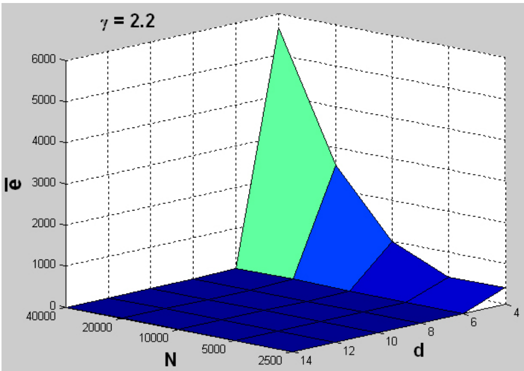

Figure 7: $\gamma = 2.2$.

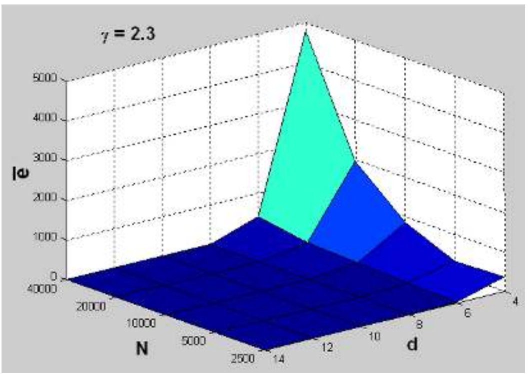

Figure 8: $\gamma = 2.3$.

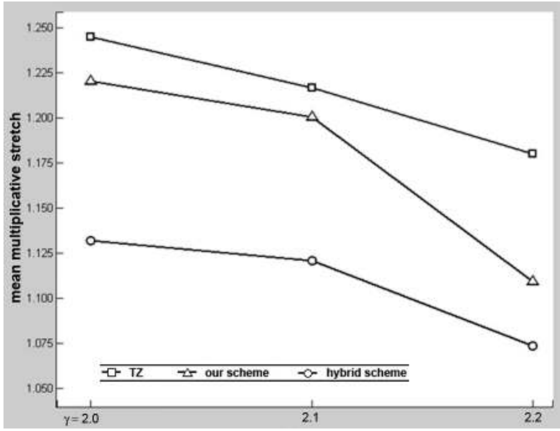

Figure 9: Observed average-case multiplicative stretch for the TZ universal scheme, the scheme presented in this paper, and the hybrid scheme mentioned in Section 3.3.

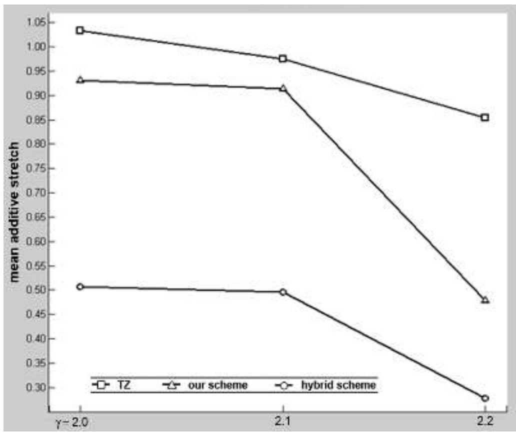

Figure 10: A comparison of average-case additive stretch for each of the three schemes.

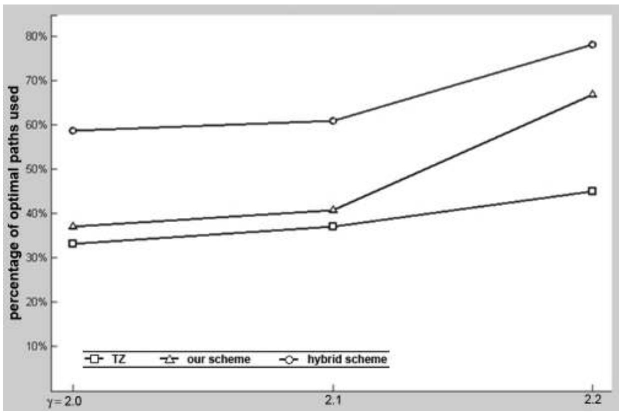

Figure 11: Percentage of optimal (i.e. shortest) paths detected by each of the three schemes in our routing simulations.

symposium on Theory of computing, pages 171–180, New York, NY, USA, 2000. ACM Press.

[3] Donald Aingworth, Chandra Chekuri, Piotr Indyk, and Rajeev Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM J. Comput.*, 28(4):1167–1181, 1999.

[4] Marta Arias, Lenore Cowen, Kofi Laing, Rajmohan Rajaraman, and Orjeta Taka. Compact routing with name independence. In *Proceedings of the 15th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 184–192, 2003.

[5] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509, 1999.

[6] Surender Baswana, Telikepalli Kavitha, Kurt Mehlhorn, and Seth Pettie. New constructions of (α, β)-spanners and purely additive spanners. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005*, pages 672–681, January 2005.

[7] Hubert T-H. Chan, Anupam Gupta, Bruce M. Maggs, and Shuheng Zhou. On hierarchical routing in doubling metrics. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 762–771, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[8] Qian Chen, Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott J. Shenker, and Walter Willinger. The origin of power laws in internet topologies revisited. In *Proc. of the IEEE INFOCOM*, 2002.

[9] Lenore J. Cowen and Christopher G. Wagner. Compact roundtrip routing in directed networks. In *Proc. 19th ACM Symp. on Principles of Distrib. Computing*, pages 51–59, 2000.

[10] Lenore Cowen. Compact routing with minimum stretch. *J. of Algorithms*, 38:170–183, 2001.

[11] Tamar Eilam, Cyril Gavoille, and David Peleg. Compact routing schemes with low stretch factor. In *17th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 11–20, 1998.

[12] Michael Elkin and David Peleg. $(1+\epsilon,\ \beta)$-spanner constructions for general graphs. *SIAM J. Comput.*, 33(3):608–631, 2004.

[13] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, pages 251–262, 1999.

[14] Georgos Siganos, Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. Power laws and the as-level internet topology. *IEEE/ACM Trans. Netw.*, 11(4):514–524, 2003.

[15] Pierre Fraigniaud and Cyril Gavoille. Local memory requirement of universal routing schemes. In *Proceedings of the 8th Annual ACM Symposium on Parallel Algorithms and Architectures*. ACM, August 1996.

[16] Pierre Fraigniaud and Cyril Gavoille. Routing in trees. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, 28^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2076 of Lecture Notes in Computer Science, pages 757–772. Springer, 2001.

[17] L. Gao. On inferring autonomous system relationships in the internet, 2000.

[18] Cyril Gavoille and Marc Gengler. Space-efficiency of routing schemes of stretch factor three. In *4th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 162–175, July 1997.

[19] Cyril Gavoille and Nicolas Hanusse. Compact routing tables for graphs of bounded genus. In *ICAL '99: Proceedings of the 26th International Colloquium on Automata, Languages and Programming*, pages 351–360, London, UK, 1999. Springer-Verlag.

[20] Jon M. Kleinberg, S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The web as a graph: Measurements, models and methods. In *Proceedings of the International Conference on Combinatorics and Computing*, July 1999.

[21] Dmitri Krioukov. Private correspondence, 2005.

[22] Dmitri Krioukov, Kevin Fall, and Xiaowei Yang. Compact routing on internet-like graphs. In *Proceedings of Infocom*, 2004.

[23] Linyuan Lu. The diameter of random massive graphs. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 912–921. Society for Industrial and Applied Mathematics, 2001.

[24] Priya Mahadevan, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Xenofontas Dimitropoulos, kc claffy, and Amin Vahdat. Lessons from three views of the internet topology, 2005.

[25] David Peleg. Proximity-preserving labeling schemes and their applications. In *WG '99: Proceedings of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 30–41, London, UK, 1999. Springer-Verlag.

[26] Liam Roditty, Mikkel Thorup, and Uri Zwick. Roundtrip spanners and roundtrip routing in directed graphs. In *Proc. 13th ACM-SIAM Symp. on Discrete Algorithms*, pages 844–851, January 2002.

[27] Mikkel Thorup and Uri Zwick. Compact routing schemes. In *Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 1–10. ACM, July 2001.

[28] Mikkel Thorup and Uri Zwick. Approximate distance oracles. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 183–192, May 2001.

[29] Walter Willinger, Ramesh Govindan, Sugih Jamin, Vern Paxson, and Scott Shenker. Scaling phenomena in the internet: Critically examining criticality. *PNAS*, 99, 2002.