Brief Announcement: Routing the Internet with Very Few Entries *

Cyril Gavoille
University of Bordeaux
LaBRI, France
gavoille@labri.fr

Christian Glacet
University of Bordeaux
LaBRI, France
glacet@labri.fr

Nicolas Hanusse
CNRS & Univ. of Bordeaux
LaBRI, France
hanusse@labri.fr

David Ilcinkas
CNRS & Univ. of Bordeaux
LaBRI, France
ilcinkas@labri.fr

ABSTRACT

This paper investigates compact routing schemes that are very efficient with respect to the memory used to store routing tables in internet-like graphs. We propose a new compact name-independent routing scheme whose theoretically proven average memory per node is upper-bounded by n^γ, with constant $\gamma < 1/2$, while the maximum memory of any node is bounded by $\sqrt{n}$ and the maximum stretch of any route is bounded by 5. These bounds are given for the Random Power Low Graphs (RPLG) and hold with high probability. Moreover, we experimentally show that our scheme is very efficient in terms of stretch and memory in internet-like graphs (CAIDA and other maps). We complete this study by comparing our analytic and experimental results to several compact routing schemes. In particular, we show that the average memory requirements is better by at least one order of magnitude than previous schemes for CAIDA maps on 16K nodes.

Keywords

compact routing; routing scheme; power-law graphs

1. INTRODUCTION

To achieve the routing task, a routing protocol typically uses *routing tables* stored at each node in order to find a path in the network. These tables are computed beforehand by what is usually called a *routing scheme*. One of the main goals in the context of routing is to reduce the storage of the routing information at each node (to allow quick routing decisions, fast updates, and scalability), while maintaining routes along paths as short as possible.

A routing scheme that guarantees a sub-linear[1] routing table size at each node is qualified to be *compact*. There is a trade-off between the route efficiency (measured in terms of *stretch*) and the memory requirements (measured by the size or the number of entries in the routing tables). An extra desirable property of a routing scheme is to use arbitrary routing addresses (say based on processor IDs or MAC addresses) and thus independent of any topological information. Such routing schemes are called *name-independent*, in contrast with *labeled routing schemes* for which nodes are labeled by poly-logarithmic size addresses that do depend on the graph and can be freely designed to help routing decisions. In practical use, a labeled routing scheme has to use a location service that maps local information to labels, which could be a bottle-neck in regards to routing scalability, mobility and multi-homing.

Trade-offs between stretch and memory for routing in arbitrary graphs are well known, and optimal name-independent algorithms exist (see for example [AGM+08]). Nevertheless for some types of routing, like routing in the AS-internet graph, optimizations can be done and trade-offs are still barely known. Indeed, this network, like many others, exhibits several structural properties that can help a lot for routing.

Routing in internet-like graphs has already received some attention in the literature. In particular [CSTW12] and [TZLL13] respectively studied labeled and name-independent compact routing scheme for internet-like graphs. They both proved that the average number of entries in the routing tables for Random Power Low Graphs (RPLG, see Fig. 1) can be significantly lower than for arbitrary graphs, and they both confirmed experimentally their analytic results on large CAIDA and "BC" maps[2].

2. RESULTS

Our contributions are the following. First we present a new stretch-5 name-independent routing scheme, that guar-

*Supported by the ANR project DISPLEXITY (ANR-11-BS02-014).

[1] W.r.t. the number of nodes in the graph.

[2] The latter maps are the benchmark graphs used in the study of [BC06]. They are based on Power Low Random Graphs (a.k.a. PLRG) a model for internet-like graphs whose analytic study is less convenient than the RPLG model.

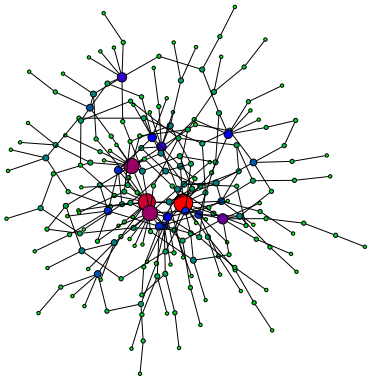

Figure 1: The largest connected component of a graph sampled from RPLG(n,τ) **for** $n = 300$ **and power-law exponent** $\tau = 2.9$**. This component has** 216 **nodes (whose size is depicted proportional to their degree),** 280 **edges, and maximum degree** 19**.**

antees, in internet-like graphs, to produce compact routing tables of very small average size at every node.

THEOREM 1. *For any n-node graph sampled from* RPLG(n,τ) *with power-low exponent $\tau \in (2,3)$, within its largest connected component, the name-independent routing scheme* CLUSTER *has w.h.p.*[3] *the following properties:*

1. *the maximal size of the routing tables is $O(\sqrt{n}\,)$;*
2. *the average size of the routing tables is*[4] *$\tilde{O}(n^{\gamma})$ with $\gamma = \frac{1}{2} \cdot \frac{\tau-2}{\tau-1}$ for $\tau \leq 2.5$, and $\gamma = \frac{(\tau-2)^2}{\tau-1}$ otherwise;*
3. *the stretch factor is at most 5;*
4. *every routing decision takes constant time; and*
5. *headers have poly-logarithmic size.*

Actually, Properties 3,4,5 hold for every connected graph.

Secondly we experimentally compare our scheme to AGMNT [AGM+08], DCR [GGHI13], HDLBR [TZLL13], and TZ$^+$ [CSTW12]. In particular, Table 1 shows that our scheme, CLUSTER, improves significantly the routing table sizes on a CAIDA map (sampled from the AS network [Cai]) and on a BC graph, even though TZ$^+$ (a specialized variant of Thorup-Zwick routing scheme) is a labeled routing scheme. Moreover, TZ$^+$ scheme makes the assumption that τ is known whereas our scheme does not need this parameter. We have provided a fully distributed implementation for the schemes DCR, HDLBR, CLUSTER, and proved that each one generates all the routing tables in $\tilde{O}(n^{3/2})$ messages. No distributed implementation within $o(n^2)$ messages is known for AGMNT.

3. OUR SCHEME

Preliminaries.

Similarly to TZ$^+$ and HDLBR, our algorithm is based on a set of *landmark nodes* positioned in the "center" of the graph. In our case, the k landmarks are composed of the highest degree node ℓ_1 plus its $k-1$ neighbors of highest degree, where $k-1 = \min\{\lceil\sqrt{n}\,\rceil, \deg(\ell_1)\}$. Moreover, every landmark node ℓ_i, $i \in \{1,\ldots,k\}$, is provided with a distinct color $c(\ell_i) \in \{1,\ldots,k\}$. Landmark nodes also share a balanced (w.h.p.) hash function h, as in [AGM+08], mapping in constant time all node identifiers to the color set $\{1,\ldots,k\}$.

[3]I.e., with probabilty at least $1 - 1/n$.

[4]The notation $\tilde{O}(f(n))$ stands for $O(f(n) \cdot \text{polylog} f(n))$.

Routing tables.

Any landmark node ℓ_i stores one entry per node v whose hash value is equal to the color of ℓ_i, namely $c(\ell_i)$. This entry corresponds to the path from ℓ_i to v in some fixed shortest-path spanning tree T rooted at ℓ_1. Each path of T can be compressed into a poly-logarithmic size entry, e.g. by using the classical labeled compact routing scheme for trees from [FG01]. This adds one entry to every node. Every landmark also stores one entry for each color c. This entry corresponds to the next-hop from ℓ_i to the landmark with color c along a shortest path in the subgraph induced by the landmarks, a diameter two subgraph. As the hash function is balanced, the number of entries for landmark routing tables is at most $k + n/k + O(1)$.

For every non-landmark node u, we define its *vicinity ball* $\mathcal{B}_u$ as the set of all nodes that are strictly closer to u than ℓ_u, where ℓ_u is a landmark closest to u. For every node v in $\mathcal{B}_u \cup N(u) \cup \{\ell_u\}$, node u stores the next-hop on a shortest path to v, where $N(u)$ denotes the neighbors of u. Thus, the number of entries for the routing table of u is at most $|\mathcal{B}_u| + \deg(u) + O(1)$. However, for each node u with no landmark neighbors, $N(u) \subseteq \mathcal{B}_u$.

Routing from u to v.

If u has an entry for v, then u can route directly to v. Otherwise u forwards the packet to ℓ_u, its closest landmark. At this point, ℓ_u computes the hash value $h(v)$ of node v and forwards the packet to the landmark ℓ_h of color $h(v)$. Finally, the information stored in the entry corresponding to v in the routing table of ℓ_h is used to route the packet to its final destination v via the tree T.

4. SKETCH OF THE PROOF

The proof of our main theorem is based on topological observations done on RPLG(n,τ) graphs [CL03]. Those graphs are defined as follows. With each node v_i, $i \in \{1,\ldots,n\}$, we assign a weight $w_i = (n/i)^{1/(\tau-1)}$. There is an edge between node v_i and v_j with probability $\min\{1, w_i w_j/\sigma\}$, where $\sigma = \sum_i w_i$.

Memory size analysis.

The first step is to show that, w.h.p., $k = \lceil\sqrt{n}\,\rceil$, i.e., the highest degree node ℓ_1 is larger than $\sqrt{n}$. This implies that landmark routing tables have at most $2\sqrt{n}$ entries, and contributes to at most $k \cdot 2\sqrt{n} = 2n$ to the total number of entries. The degree of every non-landmark u is also lower than $\sqrt{n}$. So, the maximum number of entries in a routing table is dominated by $\max_u |\mathcal{B}_u| + \sqrt{n}$. And, the average number of entries is bounded by $\frac{1}{n}\sum_u |\mathcal{B}_u| + O(1)$ since the average degree is $O(\sigma/n) = O(1)$, the main component has $\Omega(n)$ nodes, and the landmark routing tables contribute to only $2n/\Omega(n) = O(1)$ entries on the average.

Next we show that the sum of the weights, called the *volume*, of the set of nodes inside the cluster is polynomial, depends on $\tau \in (2,3)$, but is always much larger than $\sqrt{n}$. Then, we use one lemma from [CL03] which states that (w.h.p.) two sets of nodes with high volumes are adjacent

Name-indep.	RPLG(n,τ) for $\tau = 2.1$			AS graph ($n = 16\,301$)			BC graph ($n = 10$K, $\tau = 2.1$)		
	$\text{Stretch}_{\max}$	Mem_{avg}	$\text{Mem}_{\max}$	$\text{Stretch}_{\text{avg}}$	Mem_{avg}	$\text{Mem}_{\max}$	$\text{Stretch}_{\text{avg}}$	Mem_{avg}	$\text{Mem}_{\max}$
AGMNT	3	$\tilde{O}(\sqrt{n})$	$\tilde{O}(\sqrt{n})$	??	465	1 261	1.56	396	1 143
DCR	7	$\tilde{O}(\sqrt{n})$	$\tilde{O}(\sqrt{n})$	1.74	465	1 261	1.63	396	1 143
HDLBR	≥ 6	$O(\sqrt{n})$	$\Omega(n^{1/2+\varepsilon})$	1.52	106	2 324	1.24	404	1 877
CLUSTER	5	$\tilde{O}(n^{1/22})$	$O(\sqrt{n})$	1.59	4.05	415	1.75	6.47	228
Labeled									
TZ^+	5	$O(n^{1/12})$	$O(n^{1/12})$	??	??	??	1.30	55.2	??

Table 1: According to [VPSV02] the AS power-law exponent τ can be estimated to 2.1. The main component of the BC graph has 7873 nodes, whereas the AS graph is connected. It is proved in [TZLL13] that the route length for HDLBR is at most $2d + 2\delta(\tau)$ where d is the source-destination distance and $\delta(\tau)$ the inter-landmark distance. We note that, w.h.p., $\delta(2.1) > 1$, and from this observation one can derive that the maximum stretch is at least 6. We ran our own (distributed) version of HDLBR since results for AS and BC maps were not available. TZ^+ is not a name-independent routing scheme, and we have not implemented it. Thus, we have some unknown experimental values for this algorithm.

or intersect. This implies that the volume of every vicinity ball is upper bounded by a small polynomial. The last part consists in exhibiting a strong relationship between the volume and the number of nodes in the vicinity balls. We use the facts, shown in [CSTW12], that the volume of a set of nodes is likely to be equal to the sum of their degrees, and that two balls of radius r and $r+1$ do not differ too much in terms of their number of nodes.

Stretch analysis.

From the routing algorithm from u to v taken from the main connected component of G, we derive that the route length is either the distance $d = d_G(u,v)$ if $v \in \mathcal{B}_u$, or bounded by $d_G(u,\ell_u)+d_G(\ell_u,\ell_h)+d_T(\ell_h,v)$ otherwise. (Recall that the route goes first from u to ℓ_u along a shortest path in G, then to ℓ_h using the landmark subgraph, and then to v using T.) If w denotes the closest ancestor of v in T in the cluster, then the length of the route from ℓ_h to v can be bounded by $d_T(\ell_h,v) \leq d_T(\ell_h,w)+d_T(w,v) \leq d_G(\ell_v,v)+2$ by definition of ℓ_v and T. Since $d_G(\ell_u,\ell_h) \leq 2$, the route length is at most $d_G(u,\ell_u) + d_G(v,\ell_v) + 4$. Note also that $d_G(v,\ell_v) \leq d_G(v,u) + d_G(u,\ell_u) = d + d_G(u,\ell_u)$ since otherwise ℓ_u would be a closer landmark for v than ℓ_v. Assuming that $v \notin \mathcal{B}_u$ (otherwise the stretch is 1), it turns out also that $d_G(u,\ell_u) \leq d$. Overall, combining these inequalities, the route length is at most $3d+4$. However, if u is a landmark, the route length is at most $d_G(u,\ell_u)+d_G(v,\ell_v)+4 \leq d_G(u,\ell_u) + [d + d_G(u,\ell_u)] + 4 = d + 4$ since in this case $\ell_u = u$.

To summarize, either u is a landmark and the stretch is at most $(d+4)/d = 1 + 4/d \leq 5$, or u is not a landmark and the stretch is at most $(3d+4)/d = 3 + 4/d$. This is at most 5 if $d \geq 2$. On the other hand, if $d = 1$ then the stretch is only 1 since each non-landmark node u has the entries for $N(u)$. Thus, the stretch factor of our scheme is at most 5 for all connected graphs.

Optimization.

For our experimentations, we proceed to several optimizations for the CLUSTER scheme. In particular, for every non-landmark node u, we remove every entry for $w \in \mathcal{B}_u$ such that the shortest-path routing from u to w can actually be achieved by routing from u to its landmark ℓ_u. In other words, we can clean $\mathcal{B}_u$ by removing every node w such that $d_G(u,w) = 1 + d_G(u',w)$ where u' is the next-hop to reach ℓ_u from u. Another optimization addresses the landmark routing tables. For a landmark ℓ_i, the set of entries to each node v with $h(v) = c(\ell_i)$ is partionned into: 1) the entries for nodes at distance at most $r = \lceil \log n / \log\log n \rceil$ from ℓ_i; and 2) the remaining entries (for further nodes). For these latter entries we store a routing label of $\log^2 n / \log\log n = r \log n$ bits used for the routing in T as described in [FG01]. And, for the former entries we use a different routing label, still of $r \log n$ bits, composed of the first r next-hops on a shortest path from ℓ_i to v. Overall, this improves the average route length without extra cost on the number of entries or their size.

5. REFERENCES

[AGM+08] I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup. Compact name-independent routing with minimum stretch. *ACM Transactions on Algorithms*, 4(3):37, 2008.

[BC06] A. Brady and L. J. Cowen. Compact routing on power law graphs with additive stretch. In *8th ALENEX*, pp. 119–128, 2006.

[Cai] The CAIDA AS Relationships Dataset, www.caida.org/data/active/as-relationships.

[CL03] F. Chung and L. Lu. The Average Distance in a Random Graph with Given Expected Degrees. *Internet Mathematics*, 1(1):91–113, 2003.

[CSTW12] W. Chen, C. Sommer, S.-H. Teng, and Y. Wang. A compact routing scheme and approximate distance oracle for power-law graphs. *ACM Transactions on Algorithms*, 9(1):A4, 2012.

[FG01] P. Fraigniaud and C. Gavoille. Routing in trees. *28th ICALP*, vol. 2076 of LNCS, pp. 757–772, 2001.

[GGHI13] C. Gavoille, C. Glacet, N. Hanusse, and D. Ilcinkas. On the communication complexity of distributed name-independent routing schemes. In *27th DISC*, vol. 8205 of LNCS, pp. 418–432, 2013.

[TZLL13] M. Tang, G. Zhang, T. Lin, and J. Liu. HDLBR: A name-independent compact routing scheme for power-law networks. *Computer Communications*, 36(3):351–359, 2013.

[VPSV02] A. Vázquez, R. Pastor-Satorras, and A. Vespignani. Internet topology at the router and autonomous system level. *ArXiv preprint [cond-mat/0206084]*, 2002.