Router dans Internet avec quinze entrées

Cyril Gavoille, Christian Glacet, Nicolas Hanusse, David Ilcinkas

Router dans Internet avec quinze entrées †

Cyril Gavoille [1], Christian Glacet[1], Nicolas Hanusse[2], David Ilcinkas[2]

[1] *LaBRI, Université de Bordeaux, France*
[2] *LaBRI, CNRS & University de Bordeaux, France*

Cet article étudie les schémas de routage compacts qui sont très efficaces en termes de mémoires utilisées pour le stockage des tables de routage dans les graphes de type Internet. Nous proposons un nouveau schéma de routage compact avec indépendance des noms, dont la mémoire moyenne par nœud est prouvée comme étant bornée par $\sqrt{n}$, et pour lequel l'étirement maximum de toute route est au plus 7. Ces bornes sont données pour la classe RPLG (Random Power Low Graphs) et sont vraies avec forte probabilité. De plus, nous montrons expérimentalement que notre schéma est très efficace en termes d'étirement et de mémoire dans les graphes de type Internet (CAIDA et d'autres cartes). Nous complétons cette étude en comparant nos résultats analytiques et expérimentaux à plusieurs schémas de routage compact. En particulier, nous montrons que les besoins moyens en mémoire de notre schéma sont meilleurs que les schémas précédents d'au moins un ordre de grandeur pour des cartes CAIDA de 16K nœuds.

Keywords: routage compact, schéma de routage, graphes en loi de puissance

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 ‡ 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.

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 §.

†Partially supported by the ANR project DISPLEXITY (ANR-11-BS02-014).

‡. W.r.t. the number of nodes in the graph.

§. 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.

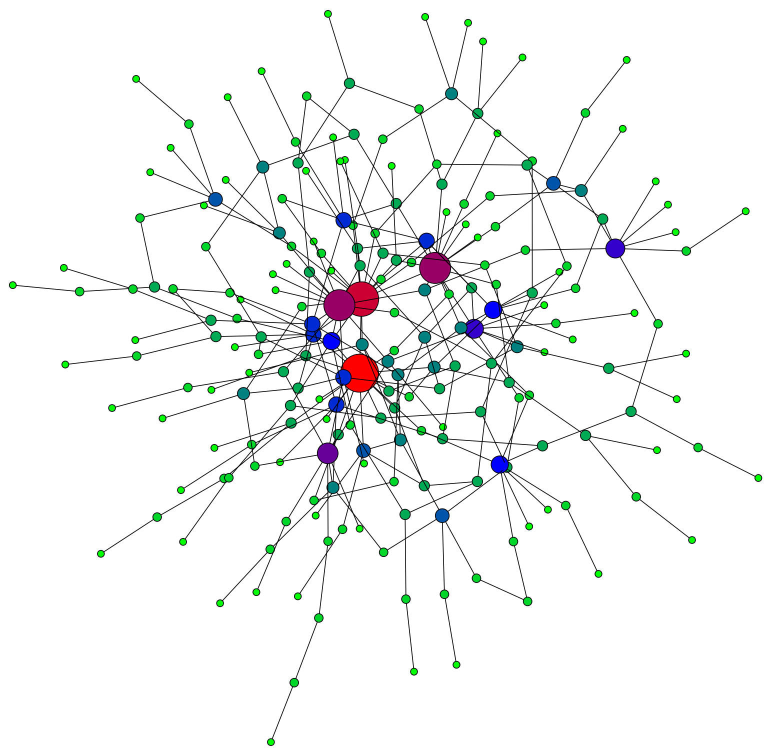

FIGURE 1: The largest connected component of a RPLG(n,t) for $n = 300$ and $t = 2.9$. The component has 216 nodes (whose size is depicted proportional to their degree) and 280 edges. The maximum degree is 19.

2 Results

Our two main contributions are the following. First we present a new name-independent routing scheme, that both guarantees, in internet-like graphs, to produce compact routing tables at every node, of very small average size, and to achieve a constant stretch factor :

> **Main Theorem.** *For any n-node graph sampled from* RPLG(n,t) *with $t \in (2,3)$, within its largest connected component, our algorithm* CLUSTER *has with high probability*[¶] *the following properties : i) the maximal size of the routing tables is $O(\sqrt{n})$; ii) the average size of the routing tables is*[‖] $\tilde{O}(n^c)$ *with $c = \frac{t-2}{2(t-1)}$ for $t \le \frac{5}{2}$, and $c = \frac{t^2-4t+4}{t-1}$ otherwise ; iii) the stretch factor is at most* 7.

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, 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 L of *landmark nodes*, that are positioned in the "center" of the graph. In our case, the set of landmarks is composed of the $k = \lceil \sqrt{n} \rceil$

¶. I.e., with probabilty at least $1 - 1/n$.

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

Name-indep	RPLG(n,t) for $t=2.1$			AS graph ($n=16\,301$)			BC graph ($n=10$K, $t=2.1$)		
	Stretch$_{\max}$	Mem$_{\text{avg}}$	Mem$_{\max}$	Stretch$_{\text{avg}}$	Mem$_{\text{avg}}$	Mem$_{\max}$	Stretch$_{\text{avg}}$	Mem$_{\text{avg}}$	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	7	$\tilde{O}(n^{1/22})$	$O(\sqrt{n})$	1.79	14.4	770	1.92	12.4	246
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 t can be estimated to 2.1. It is proved in [TZLL13] that the route length for HDLBR is at most $2(d+\delta(t))$ where d is the source-destination distance and $\delta(t)$ the inter-landmark distance. It is not difficult to see that $\delta(2.1) > 1$ w.h.p., 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.

closest nodes from the highest degree node ℓ_1, prefering nodes with highest degree at the last layer. Moreover, every landmark node ℓ_i, $i \in \{1,\dots,k\}$, is provided with a distinct color $c(\ell_i) \in \{1,\dots,k\}$. 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 set $\{1,\dots,k\}$.

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. Every landmark also stores one entry for each color c. Similarly, this entry corresponds to the path in T from ℓ_i to the landmark with color c. 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. 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 $\mathcal{B}_u \cup \{\ell_u\}$, node u stores the next-hop on a shortest path to v.

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 $h(v)$ via the tree T. 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 T.

4 Sketch of the proof

The proof of our main theorem is based on topological observations done on RPLG(n,t) graphs [CL03]. Those graphs are constructed as follows. With each node v_i, $i \in \{1,\dots,n\}$, we assign a weight $w_i = (n/i)^{1/(t-1)}$. 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 the radius of the cluster of landmarks is 1 (w.h.p.). Next we show that the sum of the weights, called the *volume*, of the set of nodes inside the cluster is polynomial, depends on $t \in (2,3)$, but is always much larger than $\sqrt{n}$. Then, we use one lemma from [CL03] which states that two sets of nodes with high volumes intersect (w.h.p.). 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. The stretch analysis is also based on the fact that the radius of the cluster is 1 (w.h.p.). 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 $d_G(u,\ell_u) + d_T(\ell_u,\ell_h) + d_T(\ell_h,v)$

otherwise. (Recall that the route goes first from u to ℓ_u along a shortest path in G, then to ℓ_h using T, and then to v using T again.) 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 $d_T(\ell_u, \ell_h) \leq 2$, 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 $3d + 4$. In other words, the stretch is at most $3 + 4/d \leq 7$.

Improved scheme. As we also show in [GGHI15], a slight modification of the scheme allows us to improve the maximum stretch to 5 with no penalties on the average stretch, and with asymptotically the same memory requirements, for the maximum and the average.

The modification is based on the observation that if one of the two nodes u, v is in the cluster, then the route length is at most $d + 4$ instead of $3d + 4$. Indeed, is v is in the cluster (i.e., $v = \ell_v$), then the route length is at most $d_G(u, \ell_u) + d_G(v, \ell_v) + 4 = d_G(u, \ell_u) + 4 \leq d + 4$. And, if u is in the cluster, then we have seen that $d_G(v, \ell_v) \leq d + d_G(u, \ell_u) = d$, and thus the route length is bounded by $d_G(u, \ell_u) + d_G(v, \ell_v) + 4 \leq d + 4$.

In particular, if every non-cluster node stores the next-hop to all its neighbors [**], the stretch becomes $(d+4)/d = 1 + 4/d \leq 5$ if u is in the cluster, it is 1 if v is a neighbor of u (thanks to the new storage in u), or it is $3 + 4/d \leq 5$ since $d \geq 2$ if the previous case does not hold.

We note that the average stretch can only be improved, that the average memory can only be worst by an additive constant number of entries (since the extra storage costs in total at most the number of edges of the whole graph, that is $O(\sigma) = O(n)$, and that the component G has $\Omega(n)$ nodes), and that the maximum number of entries remains in $O(\sqrt{n})$ since w.h.p. the highest degree of a non-cluster node is $O(\sqrt{n})$.

Références

[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 Workshop on Algorithm Engineering and Experiments (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 International Colloquium on Automata, Languages and Programming (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 International Symposium on Distributed Computing (DISC)*, vol. 8205 of LNCS, pp. 418–432, 2013.

[GGHI15] C. Gavoille, C. Glacet, N. Hanusse, and D. Ilcinkas. Brief Announcement : Routing the Internet with less than fifteen entries. In *35th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 2015.

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

**. In fact, only nodes with vincinity ball of radius 0, i.e., nodes neighboring the cluster, needs this extra storage.