EUROPHYSICS LETTERS 15 June 2006

Europhys. Lett., **74** (6), pp. 1102–1108 (2006)
DOI: 10.1209/epl/i2006-10049-1

Searching complex networks efficiently with minimal information

S. Carmi[1], R. Cohen[2] and D. Dolev[3]

[1] *Minerva Center and Department of Physics, Bar-Ilan University - Ramat-Gan, Israel*
[2] *Electrical and Computer Engineering Department, Boston University Boston, MA, USA*
[3] *School of Engineering and Computer Science, Hebrew University - Jerusalem, Israel*

received 3 March 2006; accepted in final form 13 April 2006
published online 17 May 2006

PACS. 89.20.Ff – Computer science and technology.
PACS. 89.75.Fb – Structures and organization in complex systems.
PACS. 02.50.Cw – Probability theory.

Abstract. – The ability to perform an efficient search in a complex network is of great importance in real-world systems. We suggest a method for searching for nodes when the source does not possess full information about the shortest path to the destination. By assigning new short names to nodes we are able to reduce significantly the amount of information stored at the nodes, such that the required memory needed scales only logarithmically with the network size; yet we succeed in finding the destination node through paths very close in distance to the shortest ones. The method is shown to perform particularly well on scale-free networks, exploiting its unique characteristics. This, together with other properties, makes our method extremely useful for realistic systems such as the Internet.

Introduction. – Complex systems whose interactions can be described as networks have captured much attention in recent years. In particular, understanding the structure of the Internet is of great importance, due to its rapidly increasing effect on our everyday life [1–4]. Now that significant progress was made in understanding the structural properties of complex networks, attention can be focused into developing novel methods and algorithms that exploit those unique properties for a better utilization of the network resources. One of the most important tasks in network science is searching, as follows. Assume that we are located at a certain site, or node, and we want to send some signal to some other node. We usually know the name of the destination node, but we do not necessarily know the shortest path to it. Each node on the path of the signal has to make the decision through which link to forward the signal so that it reaches the destination as quickly as possible. In the context of communication networks, the problem of searching is known as routing, routers send information packets to other routers. In social networks, people are looking to communicate with other people, who are not necessarily their close friends. For example, in the famous Milgran's experiment [5], each individual was given a name of an unknown person, and had to reach him by sending post to intermediate people. When dealing with transportation networks, one simply has to reach some place, going through a network of flights or roads.

In the case that every node is familiar with the topology of the entire network, signals can be sent through shortest paths; this requires every node to calculate the shortest path tree from it to the rest of the network. However, this is not a realistic situation in large-scale networks, such as the Internet or social networks, due to limited amount of resources (*e.g.*, memory or computation time). A trade-off must be made between the quality of the path and the requirements at the nodes. Various aspects of search with limited information have been investigated [6–10]. These include local search strategies in small-world lattices and networks with broad degree distributions, analysis of the trade-off between network congestion and search efficiency, model for the amount of information that a network needs to hold (globally) to perform shortest path searching and applications to various networks and search strategies.

For the sake of convenience of notation to this end we concentrate on the problem of routing in communication networks; this without loss of generality, as our methods can be applied in other fields as well. We will present a new method for routing where no knowledge of the location of the destination node is given. In the context of communication networks, such methods are usually known as "compact routing" schemes (see below).

Our method belongs to a class of routing schemes that are known as "labeling schemes". These work by giving new short names (labels) to the nodes in the system, such that the new names will contain information that will be useful in the process of navigation to or from the nodes. The scheme is then responsible for providing a protocol that will exploit the new names to make the best possible routing decision, *i.e.* decide, in each intermediate node of the trajectory of the information packet, what is the best link to send it through such that it reaches the destination on shortest path. To demonstrate this, let us consider the case of navigating in a city. If the names of the streets are chosen arbitrarily, it is usually difficult to seek an unknown location. But, if the streets' names are not meaningless (*e.g.* numbered as in Manhattan), navigation becomes much easier, as we know from everyday experience.

In order for a labeling scheme to obtain good results several variables should be considered:

- The *stretch* is defined as the ratio of the actual routing path to the shortest path between two given nodes. The smallest the ratio the more efficient the communication in the network.

- The *table size* is the number of entries kept in storage at each node. The smaller the table the more efficient the scheme in terms of memory requirements.

- The *label size* is the number of bits representing the name (or address) of each node. The smallest possible label size needed to distinguish between every pair of sites is logarithmic in the system size. Most efficient routing schemes use larger labels in order to represent more information about the nodes.

Note that in large-scale systems (such as the real Internet) it is almost always not possible to use a scheme in which the amount of storage needed scales like any power of the system size. It is thus desirable to design the routing scheme in a way that would require considerably smaller tables, in the cost of allowing for higher stretch and larger labels.

Graph theoreticians have been investigating compact routing for quite a while. The first work on generalized routing with a trade-off of table size *vs.* label size and stretch was given by Peleg and Upfal [11]. This scheme has later been extended by Thorup and Zwick [12] and by Cowen [13]. All those schemes suffer from having to use a rather large table whose size scales like some power of the system size. (Order of $N^{1/2}$ to ensure an upper bound of 3 for the stretch, or in general, order of $N^{\frac{2}{s+1}}$ for an odd stretch s.) A numerical study of the actual stretch for Internet-like networks is presented in [14], showing that the actual performance of

the above routing schemes, in terms of the average stretch, is much better than the worst case guarantee.

In this work we discuss a class of routing schemes with a parameter H ($1 \leq H \leq N$), which is proportional to the memory requirement at the nodes. We give arguments showing that the ratio of the average routing distance to the average shortest path is below 2 with high probability, no matter what H is. For networks with power law degree distribution (scale-free networks) such as the real Internet, the stretch is usually much lower, and we show analytically and numerically that *even* when H scales only logarithmically with the system size $H \sim \log^{\nu} N$ for $\nu \geq 0$, the actual stretch is very close to 1. Thus, a routing scheme that requires substantially smaller tables and poly-logarithmic labels (see below) may lead to a very efficient routing. When comparing properly, our scheme is found to be more efficient than previous ones; moreover, as we will show, our scheme is simpler and more intuitive (*e.g.* it does not involve randomization), and the trade-off between performance and memory requirements is controllable.

The scheme. – The proposed routing scheme consists of two stages: the preprocessing and the actual routing.

Preprocessing. The H highest degree nodes are designated as the "hubs". (Ties in the degree are broken arbitrarily.) For each site i the closest hub h_i is searched (ties are broken by degree). Designate the shortest path from site i to its hub h_i by $v_{i0}, v_{i1}, v_{i2}, \ldots, v_{i,n_i}$, where $i = v_{i,0}$ and $v_{i,n_i} = h_i$. The label for site i will be $L_i = \langle i, v_{i,1}, v_{i,2}, \ldots, v_{i,n_i-1}, h_i \rangle$. The routing table for each node in the network contains the link leading to the shortest path for each of the hubs, as well as a list of all of its immediate neighbors.

Actual routing. Assume a packet is sent from some initial node towards the destination node t. As the packet reaches some intermediate node x, it is handled by the following algorithm:

1) If $x = t$ then stop.
2) If t is a neighbor of x, then send the packet directly to t.
3) Otherwise, if $x \in L_t$, *i.e.* $x = v_{t,j}$ for some j, then move the packet to $v_{t,j-1}$.
4) Otherwise, search for h_t in the table and send the packet through the appropriate link.

Analysis. – Let us first look at the average running time. It can be easily shown that the preprocessing step can be performed in time linear in the system size (times the number of hubs H), whereas the actual routing decision can be performed in practical cases in constant time. This makes our scheme efficient by means of running times.

To analyze the performance of the scheme by means of the actual routing path length, we will consider the Configuration Model of [15] as our random network model. The networks in this model are created by the following process: given a network with N nodes, and a degree sequence $k_{i,1 \leq i \leq N}$, create a list containing k_i copies of each node i, and choose a random matching on this list to create the edges of the network. We ignore self-loops and multiple edges, which are statistically insignificant [16].

The main degree sequence we will discuss is of scale-free networks: $P(k) \sim k^{-\gamma}$, (with $k \geq k_{min}$). Networks with the above degree sequence have been extensively investigated in recent years, due to the fact that they were found to be abundant in Nature [2]. Such networks were found to describe various systems such as biological, ecological, social and technological systems. In particular, the Internet [1] and P2P networks are scale-free [17]. Another degree sequence which we will use for comparison is the one of the Erdös-Rényi (ER) random network model, $P(k) = \frac{e^{-\gamma}\gamma^k}{k!}$.

We now look at the average distance traveled by a packet relative to the average shortest path in the network. The average is taken over all pairs and all configurations of the network in the network model presented above.

We use the following lemma. Let a_1 and a_2 be nodes with respective degrees $k_{a_1} \geq k_{a_2}$, and b be any other random node. Denote by $d(a,b)$ the length of the shortest path between nodes a and b, then we claim that

$$P(d(a_1,b) \leq l) \geq P(d(a_2,b) \leq l) \tag{1}$$

for all l's.

To see this, we consider only cases in which the paths $a_1 \to b$ and $a_2 \to b$ exist (otherwise the distance is not defined). Now fix the connections in the sub-network formed by deleting a_1 and a_2 from the original network, and consider the links between this sub-network and $\{a_1, a_2\}$. Assume that p of the links lead to paths of length l, which is the length of the shortest path to b.

If the network is with high probability fully connected (as in random networks in which all degrees are at least 3 [18], and the case of the Internet), then the ratio of matchings for which $d(a_1,b) = l$ and $d(a_2,b) > l$ to those where $d(a_1,b) > l$ and $d(a_2,b) = l$ is $\binom{k_{a_1}}{p}/\binom{k_{a_2}}{p}$, and therefore the distance is a non-increasing function of the degree.

In cases where the network is not fully connected, we must condition the relevant matchings on the demand that both a_1 and a_2 are connected to b. It can be shown that also in these cases eq. (1) is valid. Therefore we conclude that, $\langle d(a,b)\rangle$, for some random node b, is a non-increasing function of k_a:

$$\forall a_1, a_2, b, \qquad k_{a_1} \leq k_{a_2} \Rightarrow \langle d(a_2,b)\rangle \leq \langle d(a_1,b)\rangle \tag{2}$$

Next we use the notation $d(a,b)$ for the length of the shortest path between nodes a and b, and $r(a,b)$ for the distance traveled by a packet sent from a to b using the above algorithm (notice that $r(a,b)$ need not be symmetric, as opposed to $d(a,b)$). We argue that in the proposed routing scheme, the expected average stretch $S \equiv \frac{\langle r(a,b)\rangle}{\langle d(a,b)\rangle} \leq 2$.

Denote the source node as s, the destination as t, the hub of t as h_t, and the lengths of the direct paths between them $d(s,t), d(s,h_t), d(t,h_t)$. By the construction of the scheme:

$$S = \frac{\langle r(s,t)\rangle}{\langle d(s,t)\rangle} \leq \frac{\langle d(s,h_t) + d(h_t,t)\rangle}{\langle d(s,t)\rangle} = \frac{\langle d(s,h_t)\rangle}{\langle d(s,t)\rangle} + \frac{\langle d(h_t,t)\rangle}{\langle d(s,t)\rangle}. \tag{3}$$

Consider first the case that the hub h_t is just a random node, call it r. Because of symmetry, there no reason why any of the distances $d(s,t)$, $d(s,r)$, $d(r,t)$ would be larger than the other, therefore on average the total routing distance $d(s,r)+d(r,t)$ is just twice the shortest distance $d(s,t)$, or the average stretch is 2.

This is true for any random node being a hub, but we are choosing the hubs as nodes with high degree. Since eq. (2) states that the average distance between a random node and a hub is smaller than the distance between two random nodes, we expect the average distances to and from the hub to be small, *i.e.* we expect $d(s,h_t) \leq d(s,t)$ and $d(h_t,t) < d(s,t)$, thus we expect that the average stretch $S \leq 2$.

(The cases in which $k_s, k_t > k_{h_t}$ are treated easily. Since h_t is the hub of t, then even if s is a hub then by the definition of the scheme h_t is closer to t than s, and $d(h_t,t) \leq d(s,t)$; if t is a hub the routing is through the shortest path by construction. Thus we can assume that s and t are not hubs and $k_s, k_t \leq k_{h_t}$.)

Note that direct application of eq. (2) is not possible since in the derivation we assumed the three nodes $\{a_1, a_2, b\}$ are fixed, while in our case rewiring might cause h_t not to be the hub closest to t anymore. Nevertheless, there is no reason to assume the inequalities will

be invalid for the reduced configuration space where we force h_t to be the hub closest to t. Paradoxically, if there is only one hub, then the three nodes are fixed and we can apply eq. (2) directly, to prove $S \leq 2$. It is however obvious, and confirmed by simulations, that increasing H would decrease the stretch.

Few more properties follow directly from the proposed scheme. a) The label size (in bits) for the proposed scheme is at most $(D+1)\log_2 N$, where D is the diameter of the network. b) The table size at every node contains $H+k$ entries, where k is the degree. c) The contents of the packet need not be changed through the routing process. Another important property is that the scheme is a shortest-path routing for a tree. This follows since in a tree there is only one path between any two nodes, so either the hub is on the path, or the destination is on the path to the hub, or there exists some node on the path to the hub which is also on the path to the destination.

For scale-free networks we can show some better bounds on the label size and the stretch. It has been shown [19,20] that with high probability the average distance between nodes scales like $\langle d \rangle \sim \log\log N$ and the diameter like $d_{max} \sim \log N$ (for $k_{min} \geq 2$ the diameter is also expected to be of order $\log\log N$). Therefore, it can be concluded that the maximum label size is of order $|L|_{max} \sim \log^2 N$ and the average label size is $\langle |L| \rangle \sim \log N \log\log N$. For scale-free networks with $\gamma < 3$, tighter bound for the stretch can be obtained. The radius of the core (the location of all high-degree nodes) is of order $\log\log N$, and almost all the mass is concentrated outside the core (see, *e.g.*, [19,21]). Now, looking at a ball around a random site with a radius a little smaller than the radius of the network, it is expected that the ball will not include the largest hub (since most sites are outside the core). Since the size of the largest hub is of order $N^{1/(\gamma-1)} \gg N^{1/2}$ [22] for $\gamma < 3$, it is expected that the ball has less than $N^{1/2}$ outgoing links (since any two balls with more than $N^{1/2}$ outgoing links are connected with high probability). Any two such balls are not expected to be connected between them, since the product of their "degree" (number of outgoing links) is less than N, so the distance between any two random sites is expected to be almost twice the radius (for a rigorous proof of this see [21]). Thus the path through the hubs is almost optimal with high probability, and the stretch between 2 randomly selected sites is expected to approach 1 for large N.

Simulation results. – To demonstrate the efficiency of the scheme, we present computer simulation results. For all networks, we use the parameters $N = 10000$, $\gamma = 2.3$, and average over many realizations. (The stretch of a network is calculated as an average over the stretch of all pairs, as in [14]). To begin with, we verify that the labels are indeed small (fig. 1(a)).

Next we have tested the scheme with the most recent representation of the Internet at the AS level [23]; the average stretch factor turned out to be as low as 1.067, with 79% of paths being shortest (as opposed to 1.09 and 71% in [14]). In fig. 1(b) we show the cumulative distribution of stretch values for routing between all pairs in a random realization of the configuration model (with power law degree distribution), for different system sizes. It can be seen that not only most of the routes are along the shortest path, but the number of exceptionally high stretches becomes more and more rare as the system grows.

Figure 2(a) shows the average stretch value as a function of the network size, compared for a few values of ν (in $H \sim \log^\nu N$) in power law networks, and for $H \sim \log^3 N$ for ER networks. It can be seen that the average stretch in the scale-free networks is significantly better than in the ER case and is virtually independent of the network size. One can also see that the stretch depends only weakly on the number of hubs; therefore, to achieve an efficient routing, one need not use too many hubs.

In fig. 2(b) we study the variation in the stretch when the parameters of the power law degree distribution are changed. We compute the stretch for $k_{min} = 1, 2, 3$ and for various

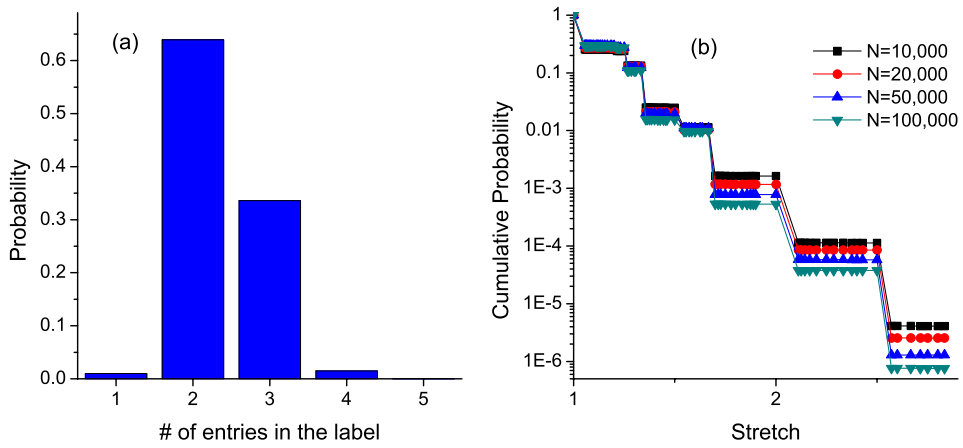

Fig. 1 – (a) Label size distribution for scale-free networks. Typical label is extremely short, what makes our scheme efficient also in terms of bandwidth utilization. (b) Stretch distribution for a scale-free network ($H \sim \log(N)$). The (inverse) cumulative probability distribution is shown, *i.e.* for a given stretch value, we plot the probability to have a larger stretch. In the case of $N = 10000$, 75% of the paths are shortest paths.

values of γ. The behavior of the stretch can be explained, as when we move to higher values of γ, the network becomes more sparse and tree-like. On the one hand, recall that the scheme is optimal for the tree structure; on the other hand, when γ increases we have less and less "real hubs", the network becomes similar to an ER network, on which the scheme performs worse, as shown above. For $k_{min} = 1$ the tree structure effect is much stronger; for $k_{min} = 3$ many loops remain thus the effect of losing the hubs is stronger; for $k_{min} = 2$ neither of the effects is more significant.

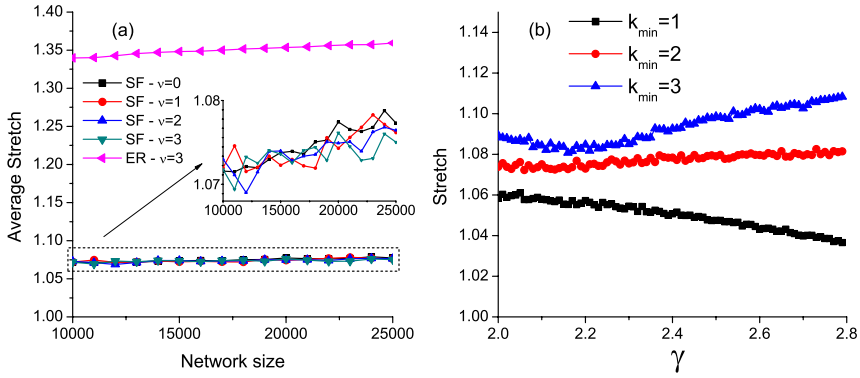

Fig. 2 – (a) Average stretch *vs.* network size, for scale-free networks with different number of hubs, $H \sim \log^{\nu} N$, $\nu = 0,\ 1,\ 2,\ 3$ and ER network ($\langle k \rangle = 7$) with $\nu = 3$. In all simulations H was scaled such that $H(N = 10000) = 100$. It can be seen that the performance of the scheme is much better for the scale-free network, with virtually no dependence in the network size and the number of hubs. (b) Average stretch *vs.* γ for a power law network with $k_{min} = 1,\ 2,\ 3$.

Summary. – In summary, we have presented an efficient method for searching, or routing, in an environment where full knowledge of the network topology is not available. Our scheme works by changing the names of the nodes to more meaningful names, that contain the path to the closest hub, where the hubs are chosen as nodes with highest degree.

In real-world systems it is important that the amount of information stored at the nodes will scale no worse than logarithmically with the system size. We have shown, numerically and analytically, that even with this constraint, our schemes perform very well. Our methods, which are simple and intuitive, can thus be considered for application in scale-free networks, such as the Internet, P2P networks, social networks or transportation networks.

* * *

We thank S. Havlin and I. Abraham for fruitful discussions. The research was supported by the Israel Internet Association, the Israel Science Foundation and the European research NEST/PATHFINDER project DYSONET 012911.

REFERENCES

[1] Faloutsos M., Faloutsos P. and Faloutsos C., *ACM SIGCOMM* (ACM Press) 1999.
[2] Albert R. and Barabási A.-L., *Rev. Mod. Phys.*, **74** (2002) 47.
[3] Pastor-Satorras R. and Vespignani A., *Evolution and Structure of the Internet* (Cambridge University Press, Cambridge) 2004.
[4] Dorogovtsev S. N. and Mendes J. F. F., *Evolution of Networks: From Biological Nets to the Internet and WWW* (Oxford University Press, Oxford) 2003.
[5] Milgram S., *Psychol. Today*, **2** (1967) 60.
[6] Kleinberg J., *Nature*, **406** (2000) 845.
[7] Adamic L. A., Lukose R. M., Huberman B. and Puniyani A. R., *Phys. Rev. E*, **64** (2001) 46135.
[8] Valverde S. and Solé R. V., *Eur. Phys. J. B*, **38** (2004) 245.
[9] Rosvall M., Grönkund A., Minnhagen P. and Sneppen K., *Phys. Rev. E*, **72** (2005) 046117.
[10] Trusina A., Rosvall M. and Sneppen K., *Phys. Rev. Lett.*, **94** (2005) 238701.
[11] Peleg D. and Upfal E., *J. ACM*, **36** (1989) 510.
[12] Thorup M. and Zwick U., *Proceedings of the Thirteenth Annual ACM Symposium on Parallel Algorithms and Architectures* 1 (ACM Press) 2001.
[13] Cowen L., *J. Algorithms*, **38** (2001) 170.
[14] Krioukov D., Fall K. and Yang X., *IEEE INFOCOM* (IEEE) 2004.
[15] Bollobas B., *Eur. J. Combinatorics*, **1** (1980) 311.
[16] Newman M. E. J., *SIAM Rev.*, **45** (2003) 167.
[17] Ripeanu M. and Foster I., *Lect. Notes Comput. Sci.*, Vol. **2429** (Springer) 2002, p. 85.
[18] Gkantsidis C., Mihail M. and Saberi A., *ACM SIGMETRICS* (ACM Press) 2003.
[19] Cohen R. and Havlin S., *Phys. Rev. Lett.*, **90** (2003) 058701.
[20] Chung F. and Lu L., *Internet Math.*, **1** (2003) 91.
[21] van der Hofstad R., Hooghiemstra G. and Znamenski D., math.PR/0502581, preprint (2005).
[22] Cohen R., Erez K., ben-Avraham D. and Havlin S., *Phys. Rev. Lett.*, **85** (2000) 4626.
[23] Shavitt Y. and Shir E., *ACM SIGCOMM Computer Commun. Rev.*, **35** (2005) 71.