

A Self-organizing Routing Scheme for Random Networks

Thomas Fuhrmann*

System Architecture Group, Universität Karlsruhe (TH), 76128 Karlsruhe, Germany
Tel. +49 721 6086730 Fax +49 721 6087664 eMail: fuhrmann@ira.uka.de

Abstract. Most routing protocols employ address aggregation to achieve scalability with respect to routing table size. But often, as networks grow in size and complexity, address aggregation fails. Other networks, e.g. sensor-actuator networks or ad-hoc networks, that are characterized by “organic growth” might not at all follow the classical hierarchical structures that are required for aggregation.

In this paper, we present a fully self-organizing routing scheme that is able to efficiently route messages in random networks with randomly assigned node addresses. The protocol combines peer-to-peer techniques with source routing and can be implemented to work with very limited resource demands. With the help of simulations we show that it nevertheless quickly converges into a globally consistent state and achieves a routing stretch of only 1.2 - 1.3 in a network with more than 10^5 randomly assigned nodes.

Keywords: **Self-organization, Peer-to-Peer, Ad-Hoc Routing**

1 The Problems of Routing

Routing is the task of picking a sequence of nodes between a packet’s source and its destination. Optimally, this sequence should denote a shortest path, but a slight deterioration is tolerable for most applications. Typically, routing is done fully distributedly in a hop by hop manner, i.e. each node decides independently of all other nodes which of its neighbors is closest to the packet’s destination and thus best suited to next route the packet. However, in order to apply this mechanism, all the nodes must have – in some sense – a complete knowledge of the network. More precisely, each node must be able to identify the next hop for *all the addresses in the entire network*.

Today, large scale networks achieve scalability by aggregating node addresses in the nodes’ routing tables. This requires the network to be so structured that node addresses and network topology match. (Theoreticians call this “labeled routing” [1].) If networks are not thoroughly planned or if the once well crafted network structure becomes outdated, aggregation fails, and, as a consequence, the lengths of the routing tables explode [5].

* This work was supported by Deutsche Forschungsgemeinschaft under grant FU448-1.

Before the advent of large-scale hierarchical networks people were used to source routing bridges that routed frames in small or medium size networks with random node address distribution. The necessity to flood the network with discovery frames, however, prevented this approach from being scalable. In other words: In unstructured networks with random node addresses classical routing mechanisms like Dijkstra and Bellman-Ford do not scale memory-wise (routing table explosion), whereas classical source routing mechanisms do not scale message-wise (flooding).

However, being able to efficiently route in networks that lack a well-crafted structure (“name independent routing”) has several advantages [3]. E.g., such networks do not require any coordinated network planning. Network growth and layman modifications cannot spoil anything. Redundancy (“multihoming”) is easily achieved since there are no hierarchical structures that one has to comply with. Moreover, being able to randomly assign addresses to nodes facilitates various applications ranging from cryptographic security to content addressing.

2 Scalable Source Routing

In this paper, we present the evaluation of a novel source routing mechanism that is both message and memory efficient. (The full protocol specification is available as a technical report [2]. The source code of our implementation is available upon request.) The core idea of our proposal is the combination of overlay routing techniques with source routing. Thereby, the knowledge about the network structure is distributed over all nodes in the network, reducing the size of the routing table in each node. Yet the knowledge is efficiently accessible by any node.

We implemented the protocol in C++ and simulated random graph networks with up to 128 000 nodes. We studied several different topologies (small-world, Erdős-Renyi, unit-disk graph) and found that our protocol is applicable in all of them. Due to their great practical relevance we discuss only small-world networks, here. This class of networks comprises e.g. the telephone network and the Internet. Since our protocol has very low memory and compute resource demands (only 4 KB per node for networks of more than 100 000 nodes), our protocol seems to be very well-suited for sensor-networks.

In our simulations we specifically focused on two aspects: (1) How quick and message efficient does the system converge into its globally consistent state? (2) What is the achieved path length of the source routes as compared to shortest paths that could be obtained by classical routing algorithms? — We also studied a third important question: (3) ‘How does the node memory size influence the routing performance?’, but found that an increase beyond 4 KB (corresponding to 255 nodes in the source route cache) per node had hardly any effect in our simulations. This is an impressive proof for the overwhelming memory efficiency of our protocol!

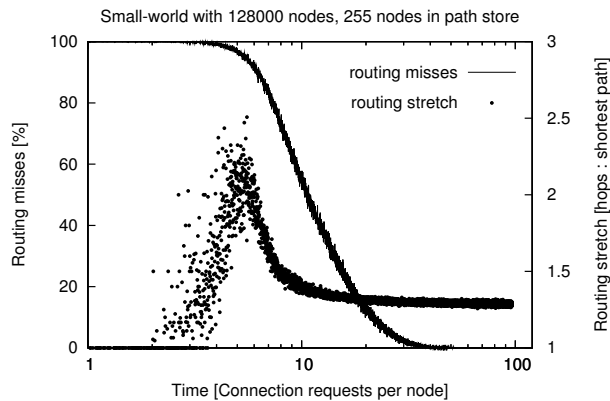


Fig. 1. Convergence and achieved routing stretch

2.1 Global consistency and achieved routing stretch

Figure 1 shows the convergence of a 128 000 node random graph network where all nodes are initialized at once. (This is a worst case scenario. If nodes are deployed successively, each node almost instantly integrates correctly.) In the simulation, each node issues connection requests for nodes that are drawn randomly (with uniform distribution) from the set of all nodes.

As can be seen, immediately after start-up, the network is inconsistent and almost all connection requests fail. Only the rare requests for addresses in the vicinity of a node can be positively answered. The resulting source routes are typically shortest paths. Hence the low routing stretch at the beginning. After each node has issued about 5–6 connection requests, the network is about to converge. Although now more and more connections can be established, the achieved routing stretch becomes worse and worse peaking at a value of 2.5. After that the knowledge about good paths builds up in the route caches and the routing stretch becomes better and better, finally stabilizing at about 1.2.

2.2 Routing Table Structure and Traffic Concentration

The memory and message efficient properties of routing in distributed hash tables like Chord [4] are based on the fact that the routing table entries of each node point to nodes in exponentially increasing distances. Fig. 2 demonstrates that this distribution is an emergent feature of our protocol. It shows the distribution of address distances between a node and all the nodes in its route cache (16 000 nodes), binned into 320 bins with exponentially growing size.

Small-world networks contain so-called *hub* nodes that serve as natural landmarks for our protocol and enable the nodes to efficiently cut down the source route paths: If two paths both happen to contain the same two hub nodes, both paths can share their knowledge about a good path between these two hub nodes. One might think that this leads to a traffic concentration at the hubs. Fig. 2

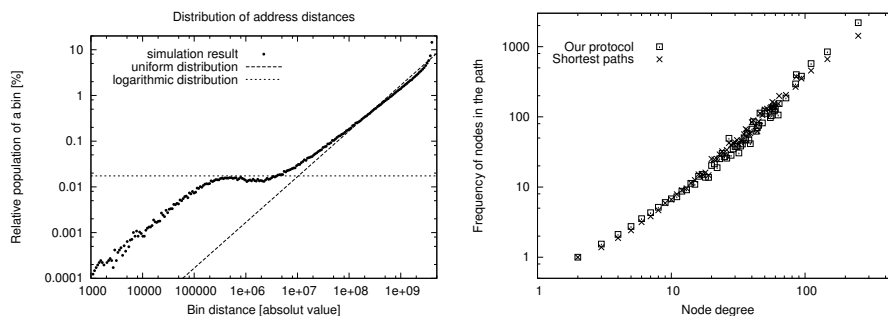


Fig. 2. Distribution of node distances (left) and node frequency in the paths (right)

show that the latter *is not the case*. Our protocol leads to the same distribution that classical shortest path algorithms achieve.

3 Conclusion and Outlook to Future Work

In this paper we presented the evaluation of a novel routing mechanism that breaks the message – memory efficiency trade-off of classical routing mechanisms. With the help of simulations we have demonstrated that the protocol is capable of achieving a routing stretch of only about 1.2 for large random networks containing nodes with random addresses. The ability to efficiently route in such environments makes our protocol especially suitable for networks that are not centrally planned or managed, and that thus do not obey strong hierarchies. Its low resource demands recommend our protocol for sensor-actuator networks where lots of small nodes need to communicate in a self-organizing manner.

Extensions of this protocol that are currently under development will improve the protocol’s capability to support node mobility and prevent malicious nodes from harming the overall routing stability.

References

1. Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, Noam Nisan, and Mikkel Thorup. Compact name-independent routing with minimum stretch. In *Proc. 16th ACM symposium on Parallelism in algorithms and architectures*, Barcelona, Spain, 2004.
2. Curt Cramer, Thomas Fuhrmann, and Kendy Kutzner. Scalable Source Routing Protocol Specification. Technical Report 2005-4, University of Karlsruhe, 2005.
3. Bryan Ford. Unmanaged Internet Protocol. *ACM SIGCOMM Computer Communications Review*, 34(1):93–98, January 2004.
4. Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proc. of ACM SIGCOMM 2001*, pages 149–160, San Diego, California, 2001.
5. Don Towsley Tian Bu, Lixin Gao. On characterizing BGP routing table growth. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 45(1), May 2004.