Brief Announcement: Compact Routing with Additive Stretch Using Distance Labelings

Arthur Brady Lenore Cowen

Tufts University Computer Science

{abrady,cowen}@cs.tufts.edu

ABSTRACT

Distance labelings – introduced as a new way to encode graph topology in a distributed fashion – have been an active area of research (see [1, 2] for details). In both exact and approximate settings, results in distance labelings and compact routing (for an introduction, esp. for definitions of **routing tables** and **headers**, see [3]) seem to go hand in hand, but so far these results have been produced separately. It was already known that graphs with constant-sized separators such as trees, outerplanar graphs, series-parallel graphs and graphs of bounded treewidth, support both exact distance labelings and optimal (additive stretch 0, multiplicative stretch 1) compact routing schemes, but there are classes of graphs known to admit exact distance labelings which do not have constant-sized separators. Our main result is to demonstrate that *every* n-vertex graph which supports an exact distance labeling with $O(l(n))$-sized labels also supports a compact routing scheme with $O(l(n) + \log^2 n)$-sized headers, $O(\sqrt{n}(l(n) + \log^2 n))$-sized routing tables, and an additive stretch of 6. Our general result produces the first known compact routing schemes for classes of graphs where no previous compact routing scheme was known, such as permutation graphs.

We note that it is possible to improve substantially on our general result for the classes of interval graphs and circular arc graphs (neither of which admits constant-sized separators). In both cases, a compact routing scheme exists with polylogarithmic headers and routing tables, and an additive stretch of 1; due to space constraints, we defer further discussion of these cases to future presentations of this work.

Categories and Subject Descriptors: G.2.2 [Graph Theory]: graph algorithms, network problems

General Terms: Algorithms.

Keywords: Compact routing.

1. GENERAL RESULT

The input to all of our algorithms is a network modeled as an n-vertex graph G, where each vertex v has been assigned a unique $\log n$-bit **ID**. A compact routing algorithm which guarantees that for all pairs v, w, the route P_{vw} taken by the algorithm satisfies $|P_{vw}| \leq k \cdot d(v, w)$, is said to have **multiplicative stretch** k; one which provides the (stronger) guarantee that $|P_{vw}| \leq d(v, w) + k$ is said to have **additive stretch** k.

Assume first that G supports an exact distance labeling with $O(l(n))$-sized labels, and that the maximum degree of any vertex is $O(\sqrt{n})$. For each v, store v's distance label and the labels of all of v's neighbors in v's routing table, and store only v's distance label in v's header. Optimal routing from v to w in this case is easy: search v's routing table for the label of some neighbor x of v for which $d(v, w) = d(v, x) + d(x, w)$, using the distance labels to compute these values, then route along (v, x), repeating until w is reached. Headers are of size $O(l(n))$; since the maximum degree of G is $O(\sqrt{n})$, routing tables are of size $O(\sqrt{n} \cdot l(n))$; and this algorithm guarantees multiplicative stretch 1 (additive stretch 0), since we always route along shortest paths in G.

We generalize this idea to all graphs which support exact distance labelings (regardless of maximum degree) by combining it with the optimal-stretch compact routing scheme for trees given in [3]. We identify a set of *landmarks* containing at most $\sqrt{n}$ vertices, all of degree $\geq \sqrt{n}$, and a set of shortest-path trees rooted at each landmark. Beginning at v, until a vertex of degree $\geq \sqrt{n}$ is encountered, we route as above. When such a vertex is encountered, we route to the closest landmark i, and then along the shortest-path tree rooted at i to w. Details of the construction of the landmark set and the distribution of storage of tree-routing information for each shortest-path tree have been omitted due to space constraints, but taken together, the artifacts of our construction incur a worst-case additive stretch of 6.

The authors gratefully acknowledge support from NSF grant CCR-0208629.

2. REFERENCES

[1] C. Gavoille, D. Peleg, S. Pérennes, and R. Raz. Distance labeling in graphs. *J. Algorithms*, 53(1):85–112, 2004.

[2] D. Peleg. Proximity-preserving labeling schemes. Technical Report CS97-23, Department of Mathematics & Computer Science, Weizmann Institute of Science, 1997.

[3] M. Thorup and U. Zwick. Compact routing schemes. In *Proc. SPAA 2001*: 1–10.

SPAA'06, July 30–August 2, 2006, Cambridge, Massachusetts, USA.
ACM 1-59593-262-3/06/0007.