Efficient message routing in planar networks — Source link

Greg N. Frederickson, Ravi Janardan

Institutions: Purdue University, University of Minnesota

Published on: 01 Aug 1989 - SIAM Journal on Computing (Society for Industrial and Applied Mathematics)

Topics: Routing table

1986

Efficient Message Routing in Planar Networks

Greg N. Frederickson
Purdue University, gnf@cs.purdue.edu

Ravi Janardan

Report Number:

86-638

EFFICIENT MESSAGE ROUTING IN PLANAR NETWORKS

Greg N. Frederickson
Ravi Janardan

CSD-TR-638
November 1986

EFFICIENT MESSAGE ROUTING IN PLANAR NETWORKS

Greg N. Frederickson *
Ravi Janardan †

Department of Computer Sciences
Purdue University
West Lafayette, Indiana 47907

CSD-TR-638
November 1986

* The research of this author was supported in part by the National Science Foundation under grant DCR–8320124, and by the Office of Naval Research on contract N 00014-86-K-0689.

† The research of this author was supported in part by the National Science Foundation under grant DCR–8320124.

Abstract

A scheme is presented for naming nodes and routing messages in a planar distributed network. For an n-node network, the scheme uses $O((1/\epsilon)\log n)$-bit names and $O(n^{1+\epsilon})$ items of routing information, where ϵ is any constant, $0 < \epsilon < 1/3$, and routes between any pair of nodes along a path which is, in worst case, at most 7 times longer than a shortest path. A simpler scheme that uses $O(\log n)$-bit names and $O(n^{4/3})$ items to realize a worst-case bound of 3 on the routings is also given.

Keywords and phrases. Distributed network, graph theory, planar graph, routing, separator, shortest paths.

1. Introduction

One of the primary functions in a distributed network is the routing of messages between pairs of nodes. Assuming that a cost, or distance, is associated with each edge, it is desirable to route along shortest paths. This can be accomplished easily by maintaining a complete routing table at each of the n nodes in the network. However, such tables are expensive for large networks, storing a total of $\Theta(n^2)$ items of routing information, where each item is a node name. Thus, recent research has focused on identifying classes of network topologies for which the shortest paths information at each node can be stored succinctly, if suitable names are assigned to the nodes. Optimal routing schemes using a total of $\Theta(n)$ items of routing information have been given for networks such as trees, unit-cost rings [SK,vLT1], complete networks, unit-cost grids [vLT2], and networks at the lower end of a hierarchy identified in [FJ1]. Unfortunately, the approach in the above research does not yield a space-efficient scheme for even such a simple class as the series-parallel networks. However, by shifting our focus to consider schemes that route along near-shortest paths, we have been able to generate space-efficient strategies for much broader classes of network topologies.

In this paper we present a near-optimal routing scheme for general planar networks. For any constant ϵ, $0 < \epsilon < 1/3$, our scheme can be set up to use $O((1/\epsilon)\log n)$-bit names and $O(n^{1+\epsilon})$ items of routing information, and route between any source-destination pair of nodes along a path that is, in worst case, at most 7 times longer than a shortest path. Our approach makes use of separator strategies [LT,M] to hierarchically decompose the network and generate names for the nodes. However, using only the Lipton-Tarjan [LT] separator algorithm the best we are able to achieve is a scheme that uses $O(n^{4/3})$ space, although with a better bound of 3 on the routings. To reduce the storage to $O(n^{1+\epsilon})$, we employ a combination of very sparse routing tables and interval routing [FJ1] to route in succession to a number of intermediate destinations carefully placed at higher levels of the decomposition. We show how Miller's [M] algorithm can be applied to generate the structured separators necessary for encoding the interval routing information.

In related work [FJ2, FJ3], we present a scheme that uses $O(cn \log n)$ items and handles any network that can be recursively decomposed by separators of size at most a constant c. Examples of such networks are the series-parallel networks, for which $c = 2$, and the k-outerplanar networks, for $k > 1$ a constant, for which $c = 2k$. Using only the names generated from a separator-based hierarchical decomposition of the network, our scheme realizes a worst-case bound of 3 on the routings. We show how to generate improved routings by succinctly encoding additional information in the node names. By augmenting the names with just $O(c \log c \log n)$ additional bits, we are able to reduce the bound to 2 for $c \leq 3$, and to a value ranging from 2 to 3 as c increases.

The decomposition technique used in [FJ2, FJ3] for constant-size separator networks does not yield a space-efficient solution for planar networks, since the latter have separators of size $O(\sqrt{n})$. Instead, we take advantage of a different approach for planar graph decomposition, as presented in [F]. This result is reviewed briefly in the next section. Sections 3 through 5 describe the $O(n^{4/3})$-space scheme, called Scheme I, while sections 6 through 8 discuss the $O(n^{1+\epsilon})$-space scheme, referred to as Scheme II.

A preliminary version of this paper appeared as part of [FJ3].

2. Division of a planar graph

We model our network with an undirected planar graph. A *planar graph* is a graph that can be embedded in the plane so that edges do not cross [H]. A *division* of a planar graph is a grouping of its nodes into subsets called *regions*. A region contains two types of nodes, namely interior nodes and boundary nodes. An *interior node* is contained in exactly one region and is adjacent only to nodes contained in the region, whereas a *boundary node* is contained in two or more regions.

For any parameter $f(n) < n$, an $f(n)$-*division* of a planar graph is a division into $\Theta(n/f(n))$ regions with a total of $O(n/\sqrt{f(n)})$ boundary nodes, where each region contains $O(f(n))$ nodes and $O(\sqrt{f(n)})$ boundary nodes. An $f(n)$-division algorithm, based on the planar separator of [LT], is given in [F].

3. Hierarchical decomposition and naming in Scheme I

The node names are generated from a multi-level division of the network G into regions. The level 0 region R_1 consists of the nodes of G, with all nodes interior. In general, the name of a region is of the form R_γ, where γ is a sequence of positive integers. Let $f(n) < n$ be a parameter to be specified later. For $i \geq 1$, let R_γ be a level $i-1$ region with n' nodes. If R_γ has a nonzero number of interior nodes, then the $f(n')$-division algorithm is applied to it to generate the level i regions $R_{\gamma 1}, R_{\gamma 2}, \ldots, R_{\gamma r}$, for some positive integer $r > 1$.

While performing the division of R_γ, we treat the boundary nodes of R_γ as boundary nodes of the resulting level i regions also. From the arguments in [F] it can be shown that, for the choice of $f(\cdot)$ to be made, the division is still an $f(\cdot)$-division. A node v that is interior to R_γ and first becomes a boundary node during its division is a *level i node*. Any other level i node generated by the division of R_γ is a *sibling* of v. A boundary node u of a level $j < i$ region to which v is interior is an *ancestor of v for level j*. We call v a *descendant* of u. Two nodes are *related* if one is an ancestor of the other or if they are siblings.

Each level i node resulting from the division of R_γ is assigned the name γ, with a distinguisher appended. This naming has the property that for unrelated nodes v and u, the length l of the longest common prefix of the distinguisher-free portions of their names is the smallest integer for which the nodes are in different level l regions. We call level l the *separating level for v and u*.

Additional information is encoded into the name of a level i node v, identifying the closest ancestor of v for each level $j < i$. An integer, called a *level j designator*, is associated with each boundary node of the level j regions that result from the division of the level $j-1$ region to which v is interior. The level j designator of the closest ancestor of v for level j is recorded in v's name, in the j^{th} field following the distinguisher.

The length of the names depends on the parameter $f(\cdot)$. We show later that for the choice of $f(\cdot)$ to be made, the names are $O(\log n)$ bits long.

4. Routing information in Scheme I

A routing table is maintained at v, giving for each related node u the name $w = next\_node_v(u)$ of the next node on a shortest (v,u)-path in G. To route to u, v sends the message to w over edge $\{v,w\}$. If w and u are unrelated, then additional information is stored at v to facilitate the routing at w. This information consists of the names a_1 and a_2 of a pair of related nodes on a shortest (v,u)-path through w, where a_1 is an ancestor of w and a_2 an ancestor of u. These names are made available to w by v in the message header, and the routing from w to u proceeds through a_1 and a_2, in that order. Let R be the region such that w and u are both in R, but are in different regions R' and R'' resulting from the division of R. Then a_1 and a_2 are respectively the first boundary nodes of R' and R'' on a shortest (v,u)-path through w. Nodes a_1 and a_2 are respectively called the next milestone and final milestone for u at v, denoted $next\_milestone_v(u)$ and $final\_milestone_v(u)$.

Figure 1 illustrates schematically a two-level division of a planar graph. A shortest (v,u)-path is shown in bold, where the neighbor w of v on this path and node u are unrelated. Both w and u are in region R_{11}, but in different regions R_{113} and R_{111} resulting from the division of R_{11}. Nodes a_1 and a_2 are the first boundary nodes of R_{113} and R_{111}, respectively, on the shortest (v,u)-path.

The following information is stored at v to enable routing to unrelated nodes. Let v be a level i node, and consider the division of the level $i-1$ region to which v is interior into level i regions. A table is stored at v, mapping the level i designator of each boundary node of the level i regions to its name. Furthermore, the name of the closest ancestor of v for each level $j < i$ is stored at v.

The amount of routing information in the network depends on $f(\cdot)$. As the following theorem shows, the appropriate choice for $f(n)$ is $n^{2/3}$.

Theorem 1. For any n-node planar graph, Scheme I uses a total of $O(n^{4/3})$ items of routing information.

Proof. We count the amount of routing information by levels. Since there are $O(n/\sqrt{f(n)})$ level 1 nodes, the level 1 nodes together maintain a total of $O((n/\sqrt{f(n)})^2)$ items of shortest paths information for siblings. As there are $\Theta(n/f(n))$ level 1 regions, each containing $O(f(n))$ nodes and $O(\sqrt{f(n)})$ boundary nodes, the level 1 nodes store a total of $O(\sqrt{f(n)}f(n)n/f(n))$, i.e., $O(n\sqrt{f(n)})$ items of shortest paths information for descendants. The size of the table of designators of a level 1 node $O(n/\sqrt{f(n)})$, so that the space used by the designator tables of all level 1 nodes is $O((n/\sqrt{f(n)})^2)$.

The descendants of the level 1 nodes store $O(n\sqrt{f(n)})$ items of shortest paths information overall for the level 1 nodes, and $O(f(n)n/f(n))$, i.e., $O(n)$ items identifying nearest level 1 nodes.

Let $S(n)$ be the total amount of information stored in an n-node network. Then, for positive constants a, b and c, we have $S(n) \leq an^2/f(n) + bn\sqrt{f(n)} + c(n/f(n))S(f(n))$, where the last term accounts for the information stored at lower levels in the decomposition. We choose $f(n) = n^{2/3}$ to make the opposing terms $an^2/f(n)$ and $bn\sqrt{f(n)}$ equal, to within a constant factor. Making an appropriate choice for the basis, we have, for positive constants d and e,

$$S(n) \leq en^{4/3}, \text{ for } n < (2c)^9,$$

$$S(n) \leq dn^{4/3} + cn^{1/3}S(n^{2/3}), \text{ for } n \geq (2c)^9.$$

Then we claim that $S(n) \leq gn^{4/3}$, where $g = \max\{e, 2d\}$. The claim can be shown by induction on n. The claim clearly holds for the basis, which is $n < (2c)^9$. For the induction step, $n \geq (2c)^9$, we require $dn^{4/3} + cn^{1/3}gn^{8/9} \leq gn^{4/3}$, i.e., $d + cgn^{-1/9} \leq g$, i.e., $d + g/2 \leq g$, which is true.

A few points are in order about our choice of constants. First, our choice of the threshold for n as $(2c)^9$ was arbitrary. A threshold of $((1+\delta)c)^9$ for any $\delta > 0$ will do. Correspondingly, $g = \max\{e, (1+1/\delta)d\}$. The second remark concerns the size of the constant e. We can bound e from above (quite generously) as follows.

Certainly, $O(n)$ is an upper bound on the number of items maintained per level by any node. Since there are $O(\log\log n)$ levels, the total space is at most $hn^2 \log\log n$ items, for some small constant h. Thus $e \leq hn^{2/3} \log\log n < h(2c)^6 \log\log(2c)^9$, since $n \leq (2c)^9$. ∎

We next show that for the above choice of $f(n)$ the node names are only $O(\log n)$ bits long. With Scheme II in mind, we prove the claim for $f(n) = n^{1-\epsilon}$, where ϵ is any constant, $0 < \epsilon < 1$.

Theorem 2. Consider the naming of the nodes of an n-node planar graph from a multi-level division, performed with respect to a parameter $f(n) = n^{1-\epsilon}$, where ϵ is any constant, $0 < \epsilon < 1$. The node names are at most $(3 + 1/\epsilon)\log n + O((1/\epsilon)\log\log n)$ bits long.

Proof. At level 0 in the decomposition there is just one region of size n and no boundary nodes. It is easy to show that the division of a level $j-1$ region into level j regions, $j \geq 1$, results in $O(n^{(1-\epsilon)^{j-1}\epsilon})$ level j regions and $O(n^{(1-\epsilon)^{j-1}(1+\epsilon)/2})$ boundary nodes of these level j regions. Furthermore, it can be established that the highest level number L in the decomposition is at most $1 + \log_{1/(1-\epsilon)} \log n = 1 + \log\log n / \log(1/(1-\epsilon))$.

For $i \geq 1$, the name of a level i node v consists of $2i$ fields, namely: i integers which constitute the name of the level $i-1$ region to which v is interior; an integer distinguisher; and $i-1$ integers, each a level j designator, $0 \leq j \leq i-1$. These fields are separated by $2i-1$ delimiters. The delimiter and the bits 0 and 1 used in the binary representation of the fields can be encoded using two bits each.

The number of bits needed to encode the region name is at most

$$\begin{aligned} &2(1+\sum_{j=1}^{i-1}\lceil \log n^{(1-\epsilon)^{j-1}\epsilon}\rceil) \\ &< 2(1+\sum_{j=1}^{i-1}(1+\log n^{(1-\epsilon)^{j-1}\epsilon})) \\ &= 2(i+(1-(1-\epsilon)^{i-1})\log n). \end{aligned}$$

The number of bits needed for the distinguisher is at most

$$\begin{aligned} &2\lceil \log n^{(1-\epsilon)^{i-1}(1+\epsilon)/2}\rceil \\ &< 2(1+\log n^{(1-\epsilon)^{i-1}(1+\epsilon)/2}) \\ &= 2(1+((1-\epsilon)^{i-1}(1+\epsilon)/2)\log n). \end{aligned}$$

The number of bits needed to encode all the designators is at most

$$\begin{aligned} &2\sum_{j=1}^{i-1}\lceil \log n^{(1-\epsilon)^{j-1}(1+\epsilon)/2}\rceil \\ &< 2\sum_{j=1}^{i-1}(1+\log n^{(1-\epsilon)^{j-1}(1+\epsilon)/2}) \\ &= 2(i-1+((1+\epsilon)/2)(1-(1-\epsilon)^{i-1})/\epsilon\ \log n). \end{aligned}$$

Finally, the delimiters can be encoded using $2(2i-1)$ bits in all.

Summing these and simplifying, the total number of bits needed to encode the name of v is at most

$$\begin{aligned} &8i-2+(2-(1-\epsilon)^{i}+(1+\epsilon)(1-(1-\epsilon)^{i-1})/\epsilon)\log n \\ &< 8i+(2+(1+\epsilon)/\epsilon)\log n \\ &< 8L+(3+1/\epsilon)\log n,\ \text{since } i \le L \\ &= (3+1/\epsilon)\log n + O((1/\epsilon)\log\log n),\ \text{noting that } \log(1/(1-\epsilon)) > \epsilon. \ \blacksquare \end{aligned}$$

5. Routing in Scheme I

A message is routed from a source s to a destination d as follows. The message header contains separate fields for the next milestone, final milestone and the destination, all initially set to d. The next and final milestone fields alone are reset, as necessary, during the routing. Let d' and d'' respectively denote the current names in the next milestone and final milestone fields. Each node v participating in the routing performs a *routing action* as follows. It determines $w = next\_node_v(d')$, resets d'' to $final\_milestone_v(d')$ and d' to $next\_milestone_v(d')$ if these entries for d' are stored at v, and then sends the message to w.

Node s searches its routing table for $d' = d$. If found, then s and d are related, and s performs a routing action. Otherwise, let l be the separating level for s and d, and $\hat{s}$ the closest ancestor of s for level l. Then s resets d' and d'' to $\hat{s}$ and performs a routing action.

Let v be any node that the message arrives at subsequently. If $v \neq d'$, then v performs a routing action.

If $v = d' \neq d''$, then v sets d' to d'' and performs a routing action.

Suppose that $v = d' = d'' \neq d$. If v and d are related, then v sets d' and d'' to d and performs a routing action. Otherwise, v must be $\hat{s}$, and $\hat{s}$ must be a level l node (Lemma 1 below). Using its table of level l designators and the l^{th} field designator in d's name, v determines the closest ancestor $\hat{d}$ of d for level l. It then sets d' and d'' to $\hat{d}$ and performs a routing action.

If $v = d' = d'' = d$, then the routing terminates.

Lemma 1. Let l be the separating level for source s and destination d, and let $\hat{s}$ be the closest ancestor of s for level l. In the routing from s to d, let v be any final milestone different from d. If v and d are unrelated, then v must be $\hat{s}$, and $\hat{s}$ must be a level l node.

Proof. Clearly, if s and d are related, then so are v and d. Thus assume that s and d are unrelated. We first show that for each v, v and d are related, except possibly when v is $\hat{s}$.

In the routing from s to $\hat{s}$, v is always $\hat{s}$, since $\hat{s}$ is an ancestor of every node in the routing. If $\hat{s}$ and d are related, the routing is from $\hat{s}$ to d. Thus every v in this routing is an ancestor of d. However, if $\hat{s}$ and d are unrelated, then the routing is from $\hat{s}$ to $\hat{d}$. Every v in this phase is an ancestor of $\hat{d}$, and hence an ancestor of d. The message eventually reaches a final milestone that is either $\hat{d}$, or an ancestor of $\hat{d}$. Thus, in the routing from this node to d, every v is an ancestor of d. Every v in this routing is an ancestor of d.

Thus, if v and d are unrelated, then v must be $\hat{s}$. Suppose that $\hat{s}$ is a level $j < l$ node. Thus $\hat{s}$ is a boundary node of the level $l-1$ region to which s is interior. But, since s and d are interior to the same level $l-1$ region, $\hat{s}$ must be an ancestor of d, a contradiction. Thus $\hat{s}$ must be a level l node. ∎

The following theorem bounds the length of the routings generated by this scheme.

Theorem 3. Let G be a planar graph. For any nodes s and d, let $\rho(s,d)$ denote the length of a shortest (s,d)-path in G, and $\hat{\rho}(s,d)$ the length of the (s,d)-path generated in Scheme I. Then the *performance bound* of Scheme I is $\hat{\rho}(s,d)/\rho(s,d) \leq 3$.

Proof. If s and d are related then the routing is along a shortest (s,d)-path. This is because every node participating in the routing performs a routing action with respect to d', which is always on a shortest (s,d)-path. Otherwise, s routes to ancestor $\hat{s}$, and if $\hat{s}$ and d are related, then $\hat{s}$ routes to d. As both routings are along shortest paths, we have

$$\begin{aligned}\hat{\rho}(s,d) &= \rho(s,\hat{s}) + \rho(\hat{s},d)\\ &\leq \rho(s,\hat{s}) + \rho(\hat{s},s) + \rho(s,d)\\ &\leq 3\rho(s,d),\ \text{as } \rho(s,\hat{s}) \leq \rho(s,d).\end{aligned}$$

If $\hat{s}$ and d are unrelated, then $\hat{s}$ routes to $\hat{d}$, where $\hat{d}$ is a sibling or an ancestor. Consider the first occasion that a final milestone $\hat{d}'$ is reached, where $\hat{d}'$ is either $\hat{d}$, or an ancestor of $\hat{d}$, and hence an ancestor of $\hat{s}$ and d. The message is routed from $\hat{d}'$ to d. The routings from $\hat{s}$ to $\hat{d}'$ and from $\hat{d}'$ to d are both along shortest paths. Thus,

$$\begin{aligned}\hat{\rho}(s,d) &= \rho(s,\hat{s}) + \rho(\hat{s},\hat{d}') + \rho(\hat{d}',d)\\ &\leq \rho(s,\hat{s}) + \rho(\hat{s},\hat{d}') + \rho(\hat{d}',\hat{d}) + \rho(\hat{d},d)\\ &= \rho(s,\hat{s}) + \rho(\hat{s},\hat{d}) + \rho(\hat{d},d),\ \text{since } \hat{d}' \text{ is on a shortest } (\hat{s},\hat{d})\text{-path}\\ &\leq \rho(s,\hat{s}) + \rho(\hat{s},s) + \rho(s,d) + \rho(d,\hat{d}) + \rho(\hat{d},d)\\ &\leq 3\rho(s,d),\ \text{as } \rho(s,\hat{s}) + \rho(\hat{d},d) \leq \rho(s,d)\ \blacksquare\end{aligned}$$

6. An improved space bound for planar networks

We now present Scheme II in which the storage is reduced to $O(n^{1+\epsilon})$ items, where ϵ is any constant, $0 < \epsilon < 1/3$. The scheme uses $O((1/\epsilon)\log n)$-bit names and has a performance bound of 7. To reduce the storage, we maintain at each node a routing table for only certain closest ancestors and descendants. However, the previous routing strategy will not work now, since the routing tables are very sparse. To overcome this problem, we introduce an additional phase in the routing, in which the message is routed to a pair of intermediate destinations carefully placed at a higher level in the decomposition. We show how to choose a good, though not necessarily optimal, path for this phase, for which the multi-interval labelling scheme from [FJ1] can be used to succinctly encode the routing information in interval form.

The network is decomposed essentially as in Scheme I. However, in order to set up the multi-interval routing information, the boundary nodes of each region must lie on one or more cycles. For a triangulated planar graph, Miller's algorithm [M] yields an $O(\sqrt{n})$-separator that is a simple cycle. The desired regions can be generated by using this, instead of the Lipton-Tarjan separator algorithm [LT], in the $f(\cdot)$-division algorithm. The graphs induced on the regions at each level are triangulated with edges of large cost, the faces of each graph are assigned zero weight (as Miller's algorithm requires that faces be weighted), and the $f(\cdot)$-algorithm is then applied to generate the regions at the next level. As in Scheme I, the boundary nodes of each region are considered to be boundary nodes of the regions resulting from its division. The nodes are named as in Scheme I.

8. Routing information in Scheme II

Let v be a level j node, $j \geq 1$. For each level $i \geq j$, shortest paths information is maintained at v for only those of its descendants for which it is the closest ancestor for level i. Let T be a tree of shortest paths from v to these descendants. Starting at v, depth-first numbers are assigned to the nodes of T, and at each node, the edge joining it to a child is labelled by a subinterval of depth-first numbers, representing all nodes in the subtree rooted at the child. A table is stored at v, mapping node names to depth-first numbers. A shortest routing from v to any node

u in T is performed by having each node on the path use the depth-first number of u, recorded in the message header by v, to choose the appropriate edge to route over.

For each level $i < j$, the closest ancestor of v for level i is identified, and the name of the parent of v in the tree rooted at that ancestor is stored at v. Thus v can perform a shortest routing to this ancestor.

For any level $i \geq j$, let R be a level i region for which v is a boundary node. Let R' be the level $i-1$ region containing R, and B the set of boundary nodes associated with the division of R'. The following information maintained at v enables it to route to the nodes in B.

A table of designators is stored at v, mapping the level i designator of each node u in B to its name.

For each u, a level number $\bar{i}$ is maintained at v, where $\bar{i} \leq i$ is the largest integer for which there is a shortest (v, u)-path in G wholly in the level $\bar{i} - 1$ region $\bar{R}$ containing R.

Furthermore, consider each u for which there is at least one (v, u)-path in G wholly in R', and let P be a least-cost such path. The name $next\_milestone_v(u)$ of the first node from B on P (in the direction from v to u) is stored at v. The routing from v to u is performed along P. Path P consists of segments, each of which is wholly in some level i region resulting from the division of R', and whose endpoints are boundary nodes of the level i region. Furthermore, each segment is a shortest such segment. For instance, the first segment has endpoints v and $next\_milestone_v(u)$, and, without loss of generality, lies wholly in region R. Each intermediate node on this segment routes to $next\_milestone_v(u)$. The routing information for this segment can be set up using the multi-interval labelling scheme from [FJ1], as follows.

In the decomposition, the boundary nodes of each region lie on cycles. Let region R have t boundary nodes lying on $p \geq 1$ cycles. Associate an integer between 1 and t, called an *interval name*, with each boundary node by proceeding around each cycle in turn, as described in [FJ1].

The following lemma shows that the routing information for the boundary nodes of R can be encoded succinctly at each node of R as subintervals of interval names labelling each incident edge.

Lemma 2. At any node w of R, the ends of all the edges incident with w can be labelled with at most $3p + degree(w) - 2$ subintervals of $[1, t]$ such that the following is true. Let z be any boundary node of R reachable from w by a path in G that is wholly contained in R. Then, the first edge on a shortest such (w, z)-path is the one whose label at w contains the interval name of z.

Proof. Consider the graph G_R defined by the nodes and edges of R. A shortest (w, z)-path in G that is wholly contained in R is a shortest (w, z)-path in G_R. The lemma then follows from Corollary 5.1 in [FJ1], since all the boundary nodes z lie on at most p faces in G_R. ∎

The edges incident with each node of R are labelled with subintervals of interval names. At boundary node v, a table mapping the names of the other boundary nodes of R to their respective interval names with respect to R is stored. The routing from v to $next\_milestone_v(u)$ is performed by having each participating node use the interval name of $next\_milestone_v(u)$, recorded in the message header by v, to choose the appropriate edge to route over.

The following theorem bounds the number of items of routing information used by Scheme II.

Theorem 4. For any n-node planar graph, Scheme II can be set up to use $O(n^{1+\epsilon})$ items, for any constant ϵ, $0 < \epsilon < 1/3$.

Proof. We count the amount of routing information by levels. The level 1 nodes claim, in the role of closest ancestors for level 1, disjoint subsets of the level $j > 1$ nodes. Thus the tables at the level 1 nodes, which map node names to depth-first search numbers, use a total of $O(n)$ space. Furthermore, the total number of subintervals of depth-first search numbers maintained for level 1 is proportional to the number of edges of G, which is $O(n)$.

Each level $j > 1$ node maintains a constant number of items for its closest ancestor for level 1. Thus the level j nodes maintain a total of $O(n)$ items about nearest ancestors for level 1.

Each boundary node of a level 1 region maintains a constant number of items (a level 1 designator, a level number, and $next\_milestone(\cdot)$) for each of the other boundary nodes. Since there are $O(n/\sqrt{f(n)})$ boundary nodes of level 1 regions, a total of $O((n/\sqrt{f(n)})^2)$, i.e., $O(n^2/f(n))$ items is stored.

The storage used by the multi-interval routing scheme for level 1 is as follows. Each boundary node of a level 1 region maintains for each of the other boundary nodes of the region, an entry in the table which maps node names to interval names. Since there are $O(\sqrt{f(n)})$ boundary nodes per level 1 region, a total of $O((\sqrt{f(n)})^2)$, i.e., $O(f(n))$ items is stored per level 1 region. Thus, as there are $\Theta(n/f(n))$ level 1 regions, a total of $O(f(n)n/f(n))$, i.e, $O(n)$ items is stored for all level 1 regions. By Lemma 3 below, $O(n)$ intervals are used to encode the multi-interval routing information.

Let $S(n)$ the total space used for an n-node network. Then, for positive constants c and d we have $S(n) \leq dn^2/f(n) + c(n/f(n))S(f(n))$. Choose $f(n) = n^{1-\epsilon}$, for any constant ϵ, $0 < \epsilon < 1/3$. With an appropriate choice of basis we then have, for some positive constant e,

$$S(n) \leq en^{1+\epsilon}, \text{ for } n < (2c)^{1/\epsilon^2},$$

$$S(n) \leq dn^{1+\epsilon} + cn^{\epsilon}S(n^{1-\epsilon}), \text{ for } n \geq (2c)^{1/\epsilon^2}.$$

We claim that $S(n) \leq gn^{1+\epsilon}$, where $g = \max\{e, 2d\}$. The proof is by induction, and is similar to that in Theorem 1. Remarks analogous to those in Theorem 1 apply here as well. ∎

The multi-interval routing scheme at level 1 enables routing between the boundary nodes of the various level 1 regions that result from the division of the level 0 region, which has n nodes. We show in the following lemma that this scheme uses a total of $O(n)$ intervals. (In general, for any level $i \geq 1$, there are a number of multi-interval routing schemes. Each scheme is associated with a level $i-1$ region and enables routing between certain boundary nodes of the level i regions resulting from the division of the level $i-1$ region. The number of intervals used for each scheme is proportional to the size of the corresponding level $i-1$ region.)

The following lemma bounds the number of intervals used by the multi-interval labelling scheme at level 1.

Lemma 3. The multi-interval labelling scheme at level 1 uses a total of $O(n)$ intervals.

Proof. We first derive an upper bound on the number of cycles on the boundaries of the level 1 regions, counting separately each occurrence of a cycle on a level 1 region boundary. In worst case the cycles are all vertex-disjoint, so that the number of cycles is one less than the number of level 1 regions, which is $\Theta(n/f(n))$. Since each cycle is on the boundary of two level 1 regions, the desired upper bound is $\Theta(n/f(n))$.

Let p_R be the number of cycles on the boundary of level 1 region R, and let w be any node of R. From Lemma 2 it follows that the total number of intervals maintained for R by all nodes w is less than

$$\begin{aligned}&\sum_{w \in R} (3p_R + degree(w)) \\ &\leq 3p_R cf(n) + 2(3cf(n) - 6),\end{aligned}$$

since R has at most $cf(n)$ nodes for some constant c, and the induced subgraph of G on R is planar. Thus the total number of intervals for all level 1 regions is less than

$$\begin{aligned}&\sum_{\text{all } R} (3p_R cf(n) + 6cf(n)) \\ &= 3cf(n) \sum_{\text{all } R} p_R + 6cf(n) \sum_{\text{all } R} 1,\end{aligned}$$

which is $O(n)$, since, from the first part of the lemma, $\sum_{\text{all } R} p_R$ is $\Theta(n/f(n))$, and since there are $\Theta(n/f(n))$ level 1 regions. ∎

9. Routing in Scheme II

The routing from s to d is as follows. Irrespective of whether or not s and d are related, the routing is always performed via the closest ancestor $\hat{s}$ of s and the closest ancestor $\hat{d}$ of d for level l, where l is the length of the longest common prefix of the distinguisher-free portions of the names of s and d. The routing from s to $\hat{s}$ is along a shortest path tree rooted at $\hat{s}$. Using its table of designators and the l^{th} field designator in d's name, $\hat{s}$ determines $\hat{d}$. Unfortunately, unlike Scheme I, it is now not possible to perform shortest paths routing from $\hat{s}$ to $\hat{d}$, since, in general, this information will not be available at $\hat{s}$. Instead the routing from $\hat{s}$ to $\hat{d}$ is performed along a near-shortest path as follows.

Let $\bar{l} \leq l$ be the level number maintained at $\hat{s}$ for $\hat{d}$. Thus there is a shortest $(\hat{s}, \hat{d})$-path wholly contained in the enveloping level $\bar{l} - 1$ region. Let $\hat{\hat{s}}$ and $\hat{\hat{d}}$ be the closest ancestors of $\hat{s}$ and $\hat{d}$ respectively, for level $\bar{l}$. If $\bar{l} = l$ then we take $\hat{\hat{s}}$ and $\hat{\hat{d}}$ to be just $\hat{s}$ and $\hat{d}$ respectively. The routing from $\hat{s}$ to $\hat{d}$ is performed in three stages, as follows. Node $\hat{s}$ records $\bar{l}$ and $\hat{d}$ in the message header and routes to $\hat{\hat{s}}$ along a shortest $(\hat{s}, \hat{\hat{s}})$-path. Using its table of designators and the $\bar{l}^{th}$ field designator in $\hat{d}$'s name, $\hat{\hat{s}}$ determines $\hat{\hat{d}}$. It then uses interval routing information to route to $\hat{\hat{d}}$ along a path P of least cost from among those that are wholly contained in the level $\bar{l} - 1$ region. Note that at least one such path exists, namely the one consisting of the shortest paths from $\hat{\hat{s}}$ to $\hat{s}$, from $\hat{s}$ to $\hat{d}$ and from $\hat{d}$ to $\hat{\hat{d}}$. Node $\hat{\hat{d}}$ then routes to $\hat{d}$ along a shortest $(\hat{\hat{d}}, \hat{d})$-path. Finally, the message is routed from $\hat{d}$ to d along a shortest $(\hat{d}, d)$-path.

An example (s, d)-routing is shown schematically in Figure 2. (For clarity, the figure is not drawn to scale and not all regions and region names are shown.) Since s and d are in different level 4 regions R_{11111} and R_{11112}, but in the same level 3 region R_{1111}, l is 4. As level 1 region R_{11} is the first enveloping region to completely contain a shortest $(\hat{s}, \hat{d})$-path, shown dashed, $\bar{l}$ is 2. The message path is shown in bold.

The following theorem establishes the performance bound of the routing.

Theorem 5. For any planar graph G, the performance bound of Scheme II satisfies $\hat{\rho}(s,d)/\rho(s,d) \leq 7$.

Proof. If $\bar{l} = l$ then $\hat{\hat{s}}$ and $\hat{\hat{d}}$ are just $\hat{s}$ and $\hat{d}$ respectively, and it follows that P is a shortest $(\hat{s},\hat{d})$-path in G. Thus

$$\begin{aligned}\hat{\rho}(s,d) &= \rho(s,\hat{s}) + \rho(\hat{s},\hat{d}) + \rho(\hat{d},d) \\ &\leq \rho(s,\hat{s}) + \rho(\hat{s},s) + \rho(s,d) + \rho(d,\hat{d}) + \rho(\hat{d},d) \\ &\leq 3\rho(s,d), \text{ as } \rho(s,\hat{s}) + \rho(d,\hat{d}) \leq \rho(s,d).\end{aligned}$$

Otherwise $\bar{l} < l$ is the highest-numbered level for which a shortest $(\hat{s}, \hat{d})$-path in G is not contained in the enveloping level $\bar{l}$ region, but is contained in the enveloping level $\bar{l}-1$ region. The path thus leaves the level $\bar{l}$ region for the first time and reenters it for the last time via two of its boundary nodes, b_1 and b_2 respectively. Thus $\rho(\hat{s},\hat{d}) \geq \rho(\hat{s},b_1) + \rho(\hat{d},b_2) \geq \rho(\hat{s},\hat{\hat{s}}) + \rho(\hat{d},\hat{\hat{d}})$, by our choice of $\hat{\hat{s}}$ and $\hat{\hat{d}}$. Let $\mid P \mid$ be the length of P. Then $\mid P \mid = \rho(\hat{\hat{s}},\hat{s}) + \rho(\hat{s},\hat{d}) + \rho(\hat{d},\hat{\hat{d}}) \leq 2\rho(\hat{s},\hat{d})$. The length of the routing from $\hat{s}$ to $\hat{d}$ is then $\rho(\hat{s},\hat{\hat{s}}) + \mid P \mid + \rho(\hat{\hat{d}},\hat{d}) \leq 3\rho(\hat{s},\hat{d})$. Thus

$$\begin{aligned}\hat{\rho}(s,d) &\leq \rho(s,\hat{s}) + 3\rho(\hat{s},\hat{d}) + \rho(\hat{d},d) \\ &\leq \rho(s,\hat{s}) + 3(\rho(\hat{s},s) + \rho(s,d) + \rho(d,\hat{d})) + \rho(\hat{d},d) \\ &\leq \rho(s,\hat{s}) + 3(2\rho(s,d)) + \rho(\hat{d},d) \\ &\leq 7\rho(s,d). \blacksquare\end{aligned}$$

References

[F] G.N. Frederickson, "Fast algorithms for shortest paths in planar graphs, with applications", CSD-TR-486 (revised), Purdue University, April 1986. Accepted for publication in *SIAM Journal on Computing.*

[FJ1] G.N. Frederickson and R. Janardan, Optimal message routing without complete routing tables, *Proceedings of the 5th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing,* Calgary, August 1986, pp. 88–97. (A revised version appears as "Designing networks with compact routing tables", October 1986.)

[FJ2] G.N. Frederickson and R. Janardan, Space-efficient message routing in *c*-decomposable networks, CSD-TR-615 (revised), Purdue University, December 1986.

[FJ3] G.N. Frederickson and R. Janardan, "Separator-based strategies for efficient message routing", *Proceedings of the 27th Annual IEEE Symposium on Foundations of Computer Science,* Toronto, October 1986, pp. 428–437.

[H] F. Harary, *Graph Theory,* Addison-Wesley, Reading MA, 1969.

[LT] R.J. Lipton and R.E. Tarjan, "A separator theorem for planar graphs", *SIAM Journal on Applied Mathematics,* Vol. 36, No. 2, April 1979, pp. 177–189.

[M] G. Miller, "Finding small simple cycle separators for 2-connected planar graphs", *Journal of Computer and System Sciences,* Vol. 32, No. 3, June 1986, pp. 265–279.

[SK] N. Santoro and R. Khatib, "Labelling and implicit routing in networks", *The Computer Journal,* Vol. 28, No. 1, February 1985, pp. 5–8.

[vLT1] J. van Leeuwen and R.B. Tan, "Routing with compact routing tables", Technical Report RUU-CS-83-16, Department of Computer Science, University of Utrecht, The Netherlands, November 1983.

[vLT2] J. van Leeuwen and R.B. Tan, "Interval routing", Technical Report RUU-CS-85-16, Department of Computer Science, University of Utrecht, The Netherlands, May 1985.

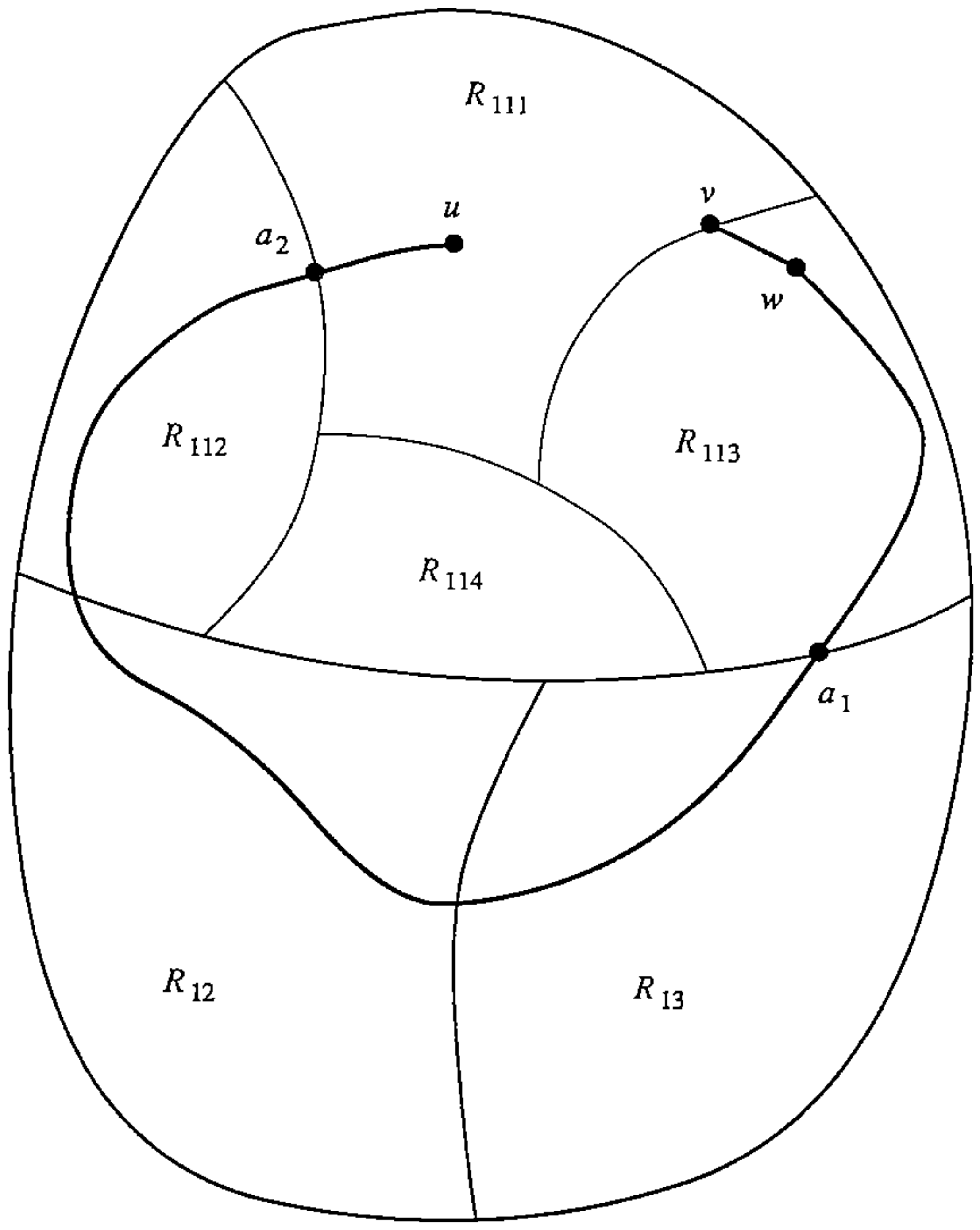

Figure 1. Illustration of the next milestone a_1 and the final milestone a_2 for u at v.

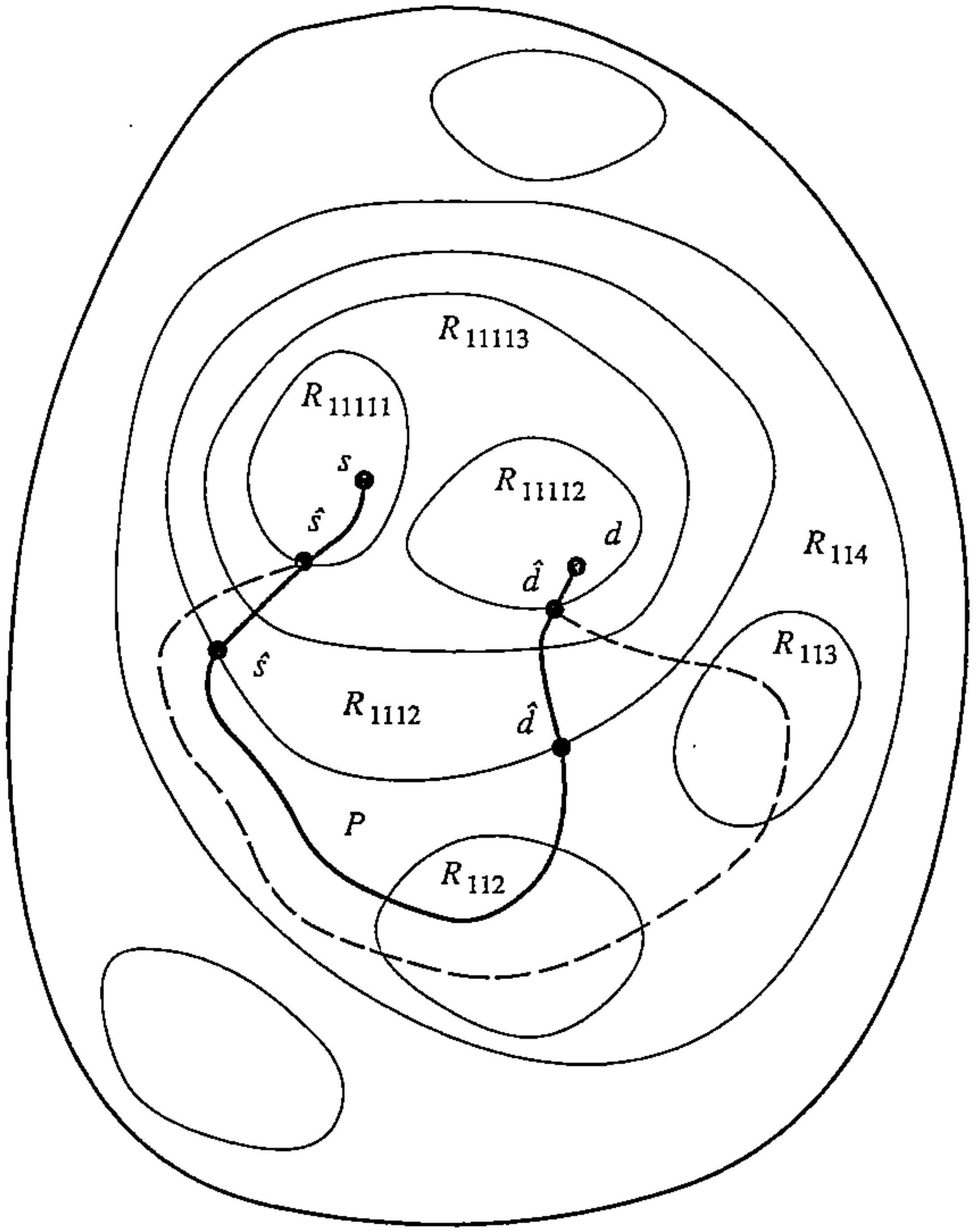

Figure 2. An example of a routing from s to d in Scheme II, shown in bold.