Linear interval routing schemes

E.M. Bakker, J. van Leeuwen, R.B. Tan

RUU-CS-91-7
February 1991

Utrecht University

Department of Computer Science

Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : ... + 31 - 30 - 531454

Linear interval routing schemes

E.M. Bakker, J. van Leeuwen, R.B. Tan

Technical Report RUU-CS-91-7
February 1991

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

ISSN:0924-3275

Linear Interval Routing *

Erwin M. Bakker , Jan van Leeuwen
Department of Computer Science, Utrecht University
P.O.Box 80.089, 3508 TB Utrecht, the Netherlands

Richard B. Tan
Department of Computer Science, Utrecht University
and
University of Science & Arts of Oklahoma
Chickasha, OK 73018, USA

Abstract

We study a variant of Interval Routing [SK85, LT86] where the routing range associated with every link is represented by a linear (i.e., contiguous) interval with no wrap around. This kind of routing schemes arises naturally in the study of dynamic Prefix Routing [BLT90]. Linear Interval Routing schemes are precisely the Prefix Routing schemes that use an alphabet of one symbol. We characterize the type of networks that admit optimum Linear Interval Routing schemes. It is shown that several well-known interconnection networks such as hypercubes, certain n-tori, and n-dimensional grids all with unit-cost links, have optimum Linear Interval Routing schemes. We also introduce the multi-label Linear Interval Routing schemes where each link may contain more than one label, and we prove several characterization results for these schemes.

1 Introduction

In communication networks routing algorithms are employed for selecting suitable routes from origin to destination nodes and ensuring that messages are correctly delivered. There is a variety of protocols for routing, most of which use extensive associated data structures known as routing tables (see e.g. [BG87]). When networks are large it becomes practical to consider methods that use compact routing

*This work was partially supported by the ESPRIT Basic Research Actions of the EC under contract no. 3075 (project ALCOM).

tables. Interval Routing schemes [SK85, LT86] were introduced as a possible technique to reduce the size of the routing tables at each node. The idea is to label each node in the network with a suitable address from the integer interval $[1, n]$, where n is the size of the network, and to associate with each link-end a label that represents an interval in $[1, n]$ (cyclic intervals allowed, i.e., intervals that wrap around over the end of the name-segment to its beginning). For every node w the intervals that are associated at w's end to the various links incident to it should be nonempty, disjoint, and together span the entire interval of possible names. Routing now proceeds as follows: At each node a message for node v is routed via the link labeled with the interval that contains $\alpha(v)$, the address of v. It is shown in [LT86] that every *fixed* graph can be assigned an Interval Routing scheme that "works", though not necessarily with optimum routes (i.e., minimum cost routes). Optimum Interval Routing schemes exist e.g. for trees, rings, grids and complete bipartite graphs [SK85, LT86], all with unit-cost links. For fixed networks with arbitrary link-costs, outerplanar graphs and c-decomposable graphs also have optimum Interval Routing schemes [FJ86, FJ88]. Furthermore, Interval Routing schemes have been adapted for dynamically growing trees [AGR89].

For dynamic networks, in which insertions and deletions of nodes and links can occur, a new type of routing scheme called a Prefix Routing scheme is introduced in [BLT90] to maintain similar advantages of routing table compactness. The idea is to assign each node an address that is a string of symbols from Σ^*, where Σ is a set of symbols of size two or more. Each link-end is also labeled with a string that represents the prefix of some addresses. A message for a destination node v is routed over the link with label l that is the maximum length label which is a prefix of $\alpha(v)$. It is shown there that every network admits a Prefix Routing scheme that "works", although again not necessarily with optimum routes. For fixed networks with arbitrary link-costs, any graph that contains no cycle of length greater than 4 has an optimum Prefix Routing scheme. It is also shown in [BLT90] that there is no containment relationship between the class of graphs with arbitrary link-costs that allow optimum Interval Routing schemes and that which allow optimum Prefix Routing schemes.

In this paper we study the case of Prefix Routing schemes in which the underlying alphabet Σ contains only one symbol. This corresponds naturally to a variant of Interval Routing, which we call Linear Interval Routing. In a Linear Interval Routing scheme each routing interval associated with a link is *linear*, i.e., does not wrap around over the end of the name-range [1,n]. We show that various well-known interconnection networks such as the hypercubes, n-dimensional grids, and certain n-dimensional tori, can all be labeled with a Linear Interval Routing scheme such that the resulting routes are optimal. We completely characterize the class of graphs that admit optimum Linear Interval Routing schemes. We also introduce the *multi-label* Linear Interval Routing scheme where each link may contain more than one label, and prove several characterization results for these schemes.

The paper is organized as follows. Section 2 contains preliminaries. Section 3 gives the characterization of graphs with optimum Linear Interval Routing schemes for dynamic link-costs and fixed link-costs. Some results on optimal Linear Interval Routing schemes for interconnection networks with unit link-costs are also presented. Section 4 contains some results on the hierarchy of multi-label Linear Interval Routing schemes.

2 Preliminaries

We assume a point-to-point communication model. Each node of a network can only communicate with its direct neighbors via the connecting links, which we assume to be bidirectional. Links may carry labels that are interpreted as costs for communicating over the link. We assume that nodes and links do not fail. The network is asynchronous and connected. When costs are considered, the cost of a link is non-negative and can be either *static* or *dynamic*. In the latter case costs can vary over time.

Definition 2.1 *A Linear Order Labeling scheme for a network G is a scheme for labeling the nodes and links incident at the nodes in G with elements from a linearly ordered set $\mathcal{L}$ with ordering relation $\preceq$ such that (i) every node gets a unique label from $\mathcal{L}$, and (ii) at every node the incident links get distinct labels from $\mathcal{L} \cup \{\infty\}$ where ∞ is a special symbol with the property that $x \prec \infty$ for every $x \in \mathcal{L}$.*

When a network G is labeled by some Linear Order Labeling scheme over a set $\mathcal{L}$, then a link incident to a node v that carries the label l at v's end implicitly defines a routing range (or interval) that consists of all names $x \in \mathcal{L}$ that are $\succeq l$ but that are not $\succeq m$ for any label $m \succeq l$ associated with another link at v. Links labeled with the special label ∞ implicitly define "empty" routing ranges.

Linear Order Labeling schemes are defined in this general way so as to leave as much freedom as possible in chosing an actual name-space. To study their combinatorial properties it is sufficient to consider name-spaces that are contiguous sets of integers only. A *Linear Interval Labeling scheme* is a Linear Order Labeling scheme over the set $\mathcal{L} = [1, n]$, for some integer n, satisfying the *Linear Interval Property*: at every node v and for every destination address there is exactly one link incident to v such that the label l that it carries at v's end defines an interval (as above) that contains the destination address. This is similar to the *Prefix Property* of a Prefix Labeling scheme over the set $\mathcal{L} = \Sigma^*$ for some singleton alphabet Σ with the natural lexicographic ordering: at every node v and for every destination address there is a unique link with a maximum length label l that is a prefix of the destination address.

The idea behind Linear Order Labeling schemes is to assign proper intervals of $\mathcal{L}$, i.e., intervals with no wrap around, to the links of a node such that the various intervals partition $\mathcal{L}$. Routing to a destination node is done via the link that is implicitly labeled with the unique interval that contains the address of the destination node. (Note that links labeled with the special label ∞ correspond to empty intervals and thus are not used for routing.) One cannot just label the links arbitrarily of course, as this may create a cycle in a routing path and the message that follows this path will then never arrive at its destination. A Linear Order Labeling scheme that routes messages correctly to their destinations is called a *valid* Linear Order Labeling scheme or simply a *Linear Order Routing scheme.* If $\mathcal{L}$ is the integer interval $[1, n]$ for some positive integer n, then we simply call it a *Linear Interval Routing Scheme* (LIRS for short).

3 Linear Interval Routing Schemes

Suppose the networks we consider are *fixed*, i.e., there are no insertions or deletions of nodes and links. The cost of the links can be either *static* (once assigned they remain fixed and never change) or *dynamic* (they can vary over time due to, say, congestion in the network or even link failures that do not disconnect the network). In this section we prove several properties of Linear Interval Routing schemes and fully characterize the class of networks that admit optimal Linear Interval Routing schemes.

3.1 Static Links

We assume that the cost of the links are arbitrary but known beforehand and thus the Linear Interval Routing scheme can be precomputed, if it exists. Even in this restricted version there are many networks that do not have an optimum LIRS.

In this section we shall make frequent use of the following notion: a *subgraph of minimum paths* of a graph is a subgraph in which the simple paths are the *only* minimum length/minimum weight paths available for the nodes in the subgraph, even in the whole network. Thus a path between nodes in the subgraph that is not contained in the subgraph of minimum paths is not minimal.

In [LT86] it is proven that every fixed graph has an Interval Labeling scheme. The next theorem shows that this is not true in the case of LIRS.

Theorem 3.1 *A tree T with unit link-costs has a LIRS if and only if it does not contain the tree T', pictured in Figure 3.1, as a subtree.*

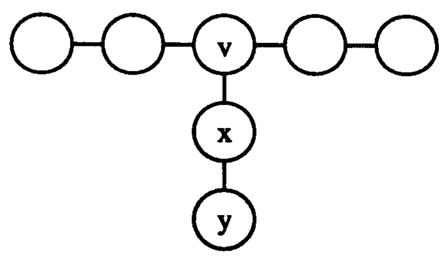


Figure 3.1

Proof. $\Rightarrow$ Assume a tree T has a LIRS and contains T' as a subtree. Let the nodes of T' be labeled with labels $u_1 < ... < u_7$. Pick the node marked with label x such that neither the labels x nor y are the highest and lowest labels u_1 or u_7. Such an x always exists since we have 3 isomorphic branches in the subgraph. Now, link (x, y) must cover the interval $[..y..]$ and link (x, v) the interval $[u_1, u_7]$. But these intervals are not disjoint; a contradiction.
$\Leftarrow$ If T does not contain T' as a subtree, then T necessarily consists of a path P with zero or more nodes of degree 1 connected to it. (Later on in this section we will recognize this as a special case of a so-called *centipede* graph.) It is easy to see that T has a LIRS. We walk along P, say from left to right, and assign ever-increasing labels to the nodes that are visited. After a label is assigned to a node of P, all its neighbors of degree 1 are labeled before proceeding to the next node on P. The links are labeled as follows. Let v and w be neighboring nodes that lie on P, and assume that $\alpha(v) < \alpha(w)$. Then $l(v, w) = [\alpha(w), n]$ and $l(w, v) = [1, \alpha(w))$. If w is a neighbor of v that does not lie on P, then $l(v, w) = [\alpha(w)]$ and $l(w, v) = [1, n]$. It is easy to verify that this constitutes a LIRS. □

Note that, if a tree T has a LIRS, then the scheme necessarily is optimum. The next lemmas show that there are many graphs that have a LIRS but that do not have an optimum LIRS.

Lemma 3.2 *Any graph that contains the following subgraph of minimum paths with unit link-costs has no optimum LIRS.*

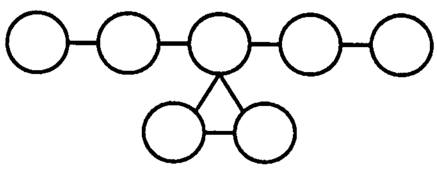


Figure 3.2

Proof. Similar to the first part of the previous proof. □

Lemma 3.3 *Any graph that contains a subgraph of minimum paths with unit link-costs that is equal to the ring C_n for some $n \geq 5$, has no optimum LIRS.*

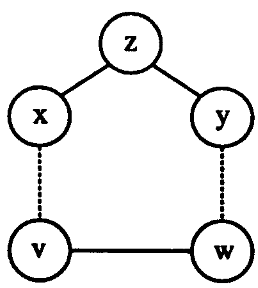


Figure 3.3

Proof. Suppose by way of contradiction that G admits an optimum LIRS. Consider a subgraph of minimum paths C_n in G, for some $n \geq 5$ (see Figure 3.3). Let the two neighbors of the node with the highest node label z be labeled x and y. W.l.o.g. we may assume that $x < y$. Pick the node v that is equidistant from nodes z and y around the ring. This is always possible if the size of the ring is odd. One link of v will cover the interval $[x, z]$ and the other link must cover the interval $[..y..]$. Hence the labels are not disjoint. So suppose n is even. Consider the node v that is equidistant from node y from either direction of the ring. Such a node always exists in an even size ring. As one link of node v must cover $[x, z]$, it follows that the neighbor of y that is not equal to z must be labeled with a label $u < x$. Consider node w. One link of w must cover $[..u..y..]$ and the other link must cover $[..x..]$. Contradiction. Thus there does not exist an optimum LIRS for a graph containing this subgraph of minimum paths. □

Lemma 3.4 *Any graph that contains the following subgraph of minimum paths has no optimum LIRS.*

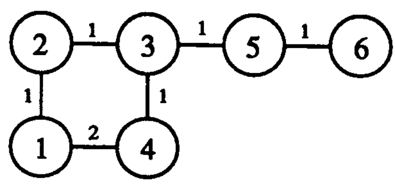


Figure 3.4

Proof. The node numbers in Figure 3.4 are the node identities and *not* the labels. Note also that link (1,4) is of cost 2 and the other links are of cost 1. Let l_{min} denote the lowest and l_{max} the highest node label used in the subgraph, respectively. If both nodes 5 and 6 or both nodes 1 and 2 are not labeled with l_{min} and l_{max}, then we can show with similar arguments as used in the first part of the proof of Theorem 3.1 that there exists no optimum LIRS. Thus we may assume that both node 3 and 4 are not labeled with l_{min} and l_{max}. Now it is clear that either l_{min} or l_{max} is assigned to node 1, for otherwise link $(4, 3)$ must cover $[l_{min}, l_{max}]$, which is an impossibility. Link $(1, 2)$ is not allowed to cover node 4. Therefore we can conclude that if l_{min} is assigned to node 1, then the label of node 4 must be smaller than the label of node 2, and if l_{max} is assigned to node 1, then the label of node 4 must be greater than the label of node 2. In both cases link $(3, 2)$ must cover node 4, which is not allowed in an optimum scheme. □

Lemma 3.5 *Any graph that contains the bipartite graph* $\mathrm{K}_{2,3}$ *shown below as a subgraph of minimum paths has no optimum LIRS.*

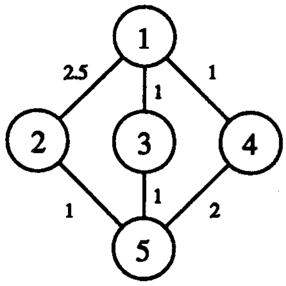


Figure 3.5

Proof. Note that link (1,2) is of cost 2.5. Consider node 2. Link $(2,5)$ must cover nodes $3,4$ and 5, and link $(2,1)$ must cover node 1. Let $l_1, ..., l_5$ be the labels of nodes $1, ..., 5$ respectively. If an optimum LIRS exists for the graph pictured in Figure 3.5, then it follows that

either (1a) $l_1 < l_3, l_4, l_5$ or (1b) $l_3, l_4, l_5 < l_1$

Considering nodes 3,5 and 1 respectively it follows that

either (2a) $l_1, l_4 < l_2, l_5$ or (2b) $l_2, l_5 < l_1, l_4$
either (3a) $l_1, l_3 < l_4$ or (3b) $l_4 < l_1, l_3$
either (4a) $l_4 < l_3, l_5$ or (4b) $l_3, l_5 < l_4$

Assume (1a). It follows that (2a) and (3a) must hold. But if (2a) holds, then (4a) must hold, and if (3a) holds, then (4b) must hold, a contradiction. Assuming (1b) it follows that (2b) and (3b) must hold. But again this yields a contradiction, as from (2b) it follows that (4b) must hold, and from (3b) it follows that (4a) must hold. □

We now state some positive results.

Definition 3.6 *Given a graph G, a G-*star *graph is the graph G with 0 or more leaves (nodes of degree* 1*) attached to the nodes of G. (See the figure below.)*

Lemma 3.7 *Any* $\mathbf{C_3}$*-star graph has an optimum LIRS.*

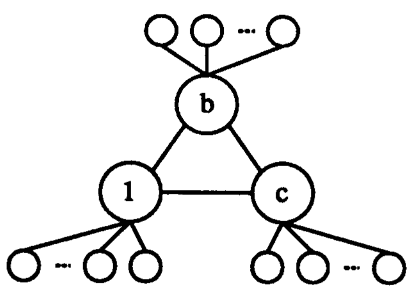


Figure 3.6

Proof. Consider a C_3-star graph (see Figure 3.6) and assume first that the shortest paths between neighboring nodes consist of the connecting link on the C_3. Label the nodes as follows. Label the nodes around the cycle as $1, b$ and c such that $b =$ "1+ the number of leaves attached to node 1" and $c =$ "b+ 1+ the number of leaves attached to node b". The labels of the leaves attached to the nodes labeled $1, b$ and c are the consecutive numbers taken from the intervals $(1, b), (b, c)$ and $(c, n]$, respectively. Here n is the number of nodes of the graph. The labels at the links are assigned as follows. The links *from* the leaves are labeled always by the interval containing all the address labels and the link *to* each leaf is labeled by the label of the leaf. For each cycle-node x, label each cycle-link with the name of the adjacent cycle-node label. This gives a valid LIRS for the graph, which is seen to give optimal routes.

Next consider the case that there are two neighboring nodes on the cycle whose shortest connecting path runs via the third node instead of via the directly connecting link. It is easily seen that this link is now effectively *forbidden* for all shortest paths between nodes in the graph. Then label the nodes incident to the forbidden link with labels 1 and c, and label the other cycle-node with b. Label the leaves as above. Label link (a, b) with b, link (b, a) with 1, link (b, c) with c, and link (c, b) with 1. It is easy to see that this is a valid LIRS. □

Lemma 3.8 *Any* K_4*-star graph that does not contain the tree* T' *pictured in Figure 3.1 as a subgraph of minimum paths has an optimum LIRS.*

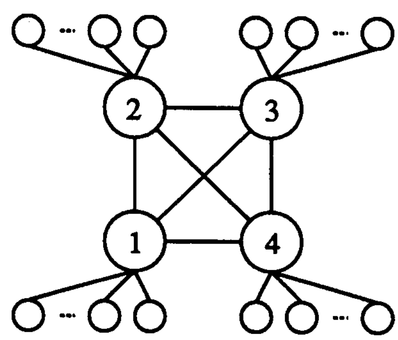


Figure 3.7

Proof. Consider a K_4-star, and its "K_4" in particular. Construct a set of shortest paths P such that every subpath of a path in P is also in P and P has a unique shortest path between each node in K_4. Furthermore the shortest path from a node u to node v in P is the same as the shortest path from node v to node u, even though there may be another shortest path.

There are 6 different cases according to the cost of the links. In the following, we will ignore the leaves, as they can easily be labeled as in Lemma 3.7. Thus we will only be concerned with labeling the nodes of K_4 with labels a, b, c, d with $a < b < c < d$ such that the resulting scheme is optimal and linear.

Case 1.

The subgraph spanned by the paths in P is a chain. Label the nodes by a, b, c, d around the chain, starting with one of the end nodes of this chain. The links are then labeled as in Figure 3.8. The scheme is linear and optimal.

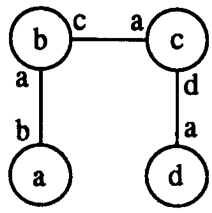


Figure 3.8

Case 2.
The subgraph spanned by the paths in P is a cycle. In this case every link occurs at least once in the set P. It is easy to see that it is impossible that every link occurs more than once in P, in the sense that it is used in the shortest path between more than one pair of nodes. Thus there is a link that is used only by the shortest path between its adjacent nodes and, furthermore, this link must be unique. Call this link the *prohibitive* link. We can now label the nodes cyclically starting at one of the nodes adjacent to this link, as in Figure 3.9. (In this figure it is assumed that (a, d) is the prohibitive link.) The scheme is linear and optimal.

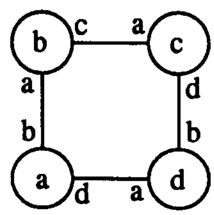


Figure 3.9

Case 3.
The subgraph spanned by the paths in P is a triangle with a branch. We can label the nodes cyclically starting with the branch-node. The links are then labeled as in Figure 3.10. The scheme is linear and optimal.

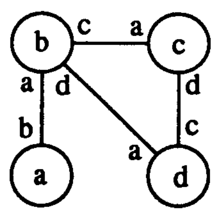


Figure 3.10

Case 4.
The subgraph spanned by the paths in P is a square with a diagonal. Clearly the

diagonal link must be the shortest path between the two diagonal nodes. The two non-diagonal nodes in P must have a shortest path through one of the diagonal nodes. Label one of them with a and then follow the link on the shortest path to the other non-diagonal node and label all the other nodes cyclically with b, c, d. The links are labeled as in Figure 3.11 to obtain an optimum LIRS.

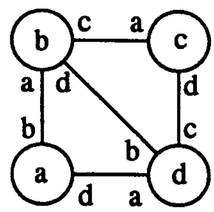


Figure 3.11

Case 5.
The subgraph spanned by the paths in P uses all links. Label the nodes in any manner with a, b, c, d and each link (x, y) with the label y to obtain an optimum LIRS.
Case 6.
The subgraph spanned by the paths in P is a $\mathbf{K_1}$*-star.* If in this case all the leaves in the subgraph have one or more leaves attached to it in the original $\mathbf{K_4}$-star, then the graph contains the tree T' as a subgraph of minimum paths and hence does not have an optimum LIRS. Otherwise there exists at least one leaf in the subgraph spanned by P that does not have leaves attached to it in the $\mathbf{K_4}$-star. Label this node with c and the other two leaves with a and d, respectively. The links are labeled as in Figure 3.12 to obtain an optimum LIRS.

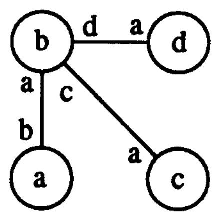


Figure 3.12

Every possible configuration must be isomorphic to one of the above cases. This shows that every $\mathbf{K_4}$-star graph that does not contain T' as a subgraph of minimum paths has an optimum LIRS. □

Definition 3.9 *A* segment *is either a* $\mathbf{C_3}$*-star graph with leaves attached to only two of the cycle nodes or a* $\mathbf{C_2}$*-star graph (cf. Figure 3.13). The two nodes of the segment with leaves attached to it are called the* head *and the* tail *of the segment, respectively.*

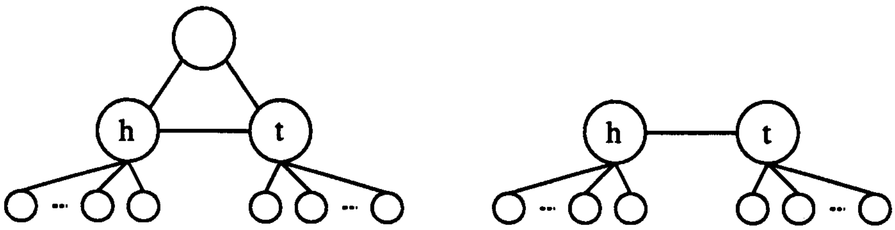


Figure 3.13

Lemma 3.10 *Every segment has an optimum LIRS.*

Proof. Similar to the proof for $\mathbf{C_3}$-star graphs. □

Definition 3.11 *A* centipede *is either a segment or a centipede joined with a segment. By joining we mean that the head of the centipede is identified with the tail of the segment that is "attached" to it, i.e., all the neighbors of these two nodes now become neighbors of the one new node. The head of the joined segment becomes the head of the new centipede (see Figure 3.14).*

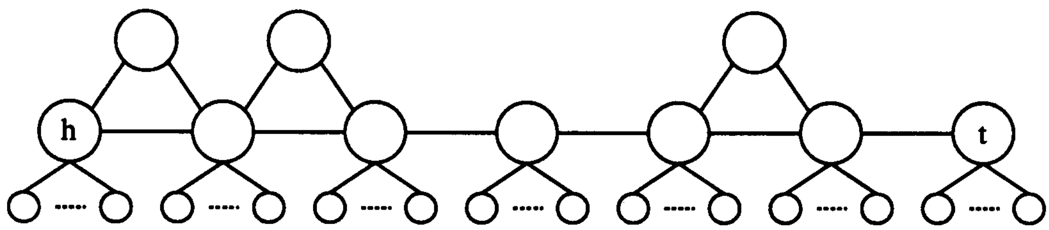


Figure 3.14

Lemma 3.12 *Every centipede has an optimum LIRS.*

Proof. Walk along the centipede from its head to its tail, and assign ever-increasing labels to the nodes that are visited. If a label is assigned to the head of a segment, we first assign labels to the leaves connected to it and assign a label to the other cycle-node (if it exists) before proceeding to the tail of the segment. It is straightforward to assign labels to the links of the centipede such that an optimum LIRS is obtained. □

We can now give a characterization of the type of graphs that allow an optimum LIRS with arbitrary fixed-cost links.

Theorem 3.13 *A graph G has an optimum LIRS for every assignment of fixed link-costs if and only if it is a centipede, a $\mathbf{C_3}$-star graph or a $\mathbf{K_4}$-star graph that does not contain the tree T' pictured in Figure 3.1 as a subgraph of minimum paths.*

Proof. From Lemma 3.3, we know that there is no optimum LIRS for graphs that contain a subgraph of minimum paths that is equal to a cycle of length 5 or more. If a graph contains more than 5 nodes and a cycle of length 4, and is not a $\mathbf{K_4}$-star, then costs can be assigned to links to obtain a subgraph of minimum paths as in Lemma 3.4. Thus these graphs do not have an optimum LIRS for all cost-assignments. Graphs of 5 nodes that contain a cycle of length 4 and are not a $\mathbf{K_4}$-star must be either $\mathbf{K_{2,3}}$ or contain a pentagon (i.e., the graph that consists of a cycle of length 4 where exactly two neighboring nodes are connected to a fifth node), neither of which has an optimum LIRS. Thus the only graphs which contain a cycle of length 4 and have an optimum LIRS are precisely the $\mathbf{K_4}$-star graphs that do not contain T' as a subgraph of minimum paths. It is easy to see that the only graphs that do not contain the graphs of Theorem 3.1 and Lemma 3.2 and which only contain cycles of length 3 are precisely the $\mathbf{C_3}$-stars and the centipedes. □

3.2 Dynamic Links

Suppose now that the cost of the links in a given network can vary over time. We are interested in graphs that have a LIRS that remains optimal despite the link-cost changes. The class of graphs that have an optimum LIRS in this case is slightly smaller.

Lemma 3.14 *Any graph with dynamic link-costs that contains a cycle of length 4 has no optimum LIRS.*

Proof. Consider a graph that contains a cycle of length 4. Suppose it has a LIRS that is optimal for all cost-assignments. Let the costs of the links not in the cycle be so large that the shortest path between any two cycle nodes must run on the cycle. Let the cycle of length 4 have unit link-costs. Let the nodes be labeled by a, b, c, d, with $a < b < c < d$. For the labeling of the nodes three cases can arise. Suppose in the LIRS the nodes are labeled cyclically by a, b, c, d, starting with (say) node 1 and continuing to node 4. Now let the cost of link (2,3) go up to 2 while the other links remain at cost 1. Then link (2,1) must contain the interval $[a, d]$ while link (2,3) must contain [..c..], a contradiction. Next, suppose the nodes are cyclically labeled with a, c, b, d, starting with node 1. Increase the cost of link (1,2) to 2 while the other links remain at cost 1. Then link (3,4) must contain the interval $[a, d]$ while link (3,2) must contain [..c..], a contradiction. Finally, suppose the nodes are cyclically labeled with a, b, d, c. Now increase the cost of link (3,4) to 2 and leave the rest of the links at cost 1. Then link (1,2) covers the interval $[b, d]$ while link (1,4) must cover [..c..], again a contradiction. □

Lemma 3.15 *Let G be a $\mathbf{C_3}$-star graph with dynamic link-costs. If every cycle-node has one or more leaves, i.e., if G is not a centipede, then G has no optimum LIRS.*

Proof. Let a, b, c be the labels of the cycle-nodes, where $a < b < c$. Let A, B, C be the sets of labels that are assigned to the leaves of the cycle-nodes labeled a, b, c, respectively. If we assign cost 1 to all links of G, it easily follows that for all nodes

$a' \in A$, $b' \in B$, and $c' \in C$: $a' < b' < c'$. If we assign costs to the links of G such that link (a, b) becomes forbidden, then link (b, c) must cover all the labels of A and C; an impossibility. □

Theorem 3.16 *A graph G with dynamic link-costs has an optimum LIRS if and only if G is a centipede.*

Proof. Lemma 3.14 rules out any graph which contains a cycle of length 4. From Section 3.1 we know that this leaves only the centipedes and $\mathbf{C_3}$-star graphs. Lemma 3.15 eliminates the $\mathbf{C_3}$-star graphs that are not centipedes. □

3.3 Unit-Cost Links

In many networks, such as processor interconnection networks, it is natural to assume that the costs of the links do not vary over time and that, furthermore, the links are all of uniform, say unit cost. The results of Theorem 3.1 and Lemmas 3.2 and 3.3 apply to networks with unit link-costs also. However Lemmas 3.4 and 3.5 are no longer true for networks with unit link-costs. It is also fairly easy to show the following results.

Lemma 3.17 *Any graph that contains one of the following graphs as subgraph of minimum paths with unit-cost links has no optimum LIRS.*

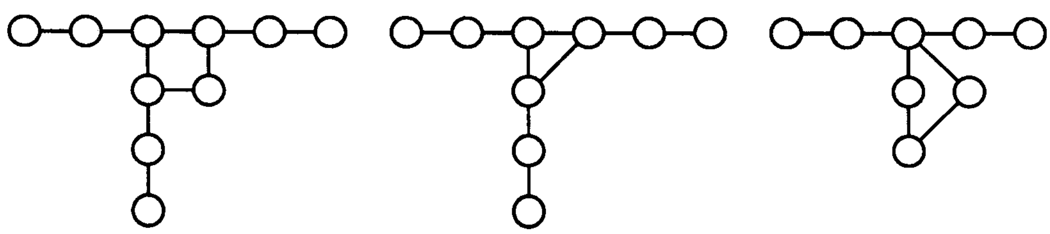
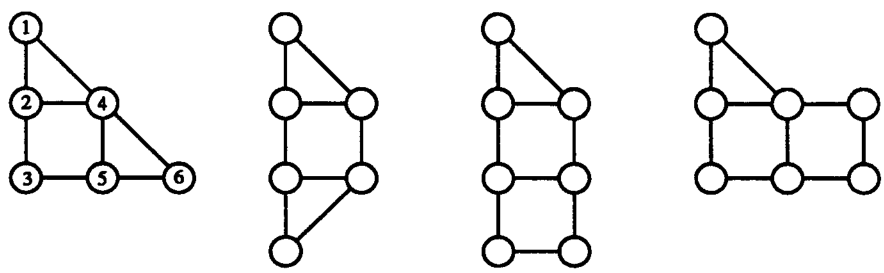


Figure 3.15

Proof. Similar to the first part of the proof of Theorem 3.1. □

Lemma 3.18 *Any graph that contains one of the following graphs as a subgraph of minimum paths with unit-cost links has no optimum LIRS.*



a) b) c) d)

Figure 3.16

Proof. We only give the proof for the subgraph depicted in Figure 3.16a. Assume that node i is labeled with label l_i for every $i \in \{1, ..., 6\}$. Consider node 1. Link (1,2) must cover the labels l_2 and l_3 and link (1,4) must cover the labels l_4, l_5 and l_6, i.e., either $l_2, l_3 < l_4, l_5, l_6$ or $l_4, l_5, l_6 < l_2, l_3$. Similarly, if we consider node 6, link (6,5) must cover the labels l_3 and l_5 and link (6,4) must cover the labels l_1, l_2 and l_4, i.e., either $l_3, l_5 < l_1, l_2, l_4$ or $l_1, l_2, l_4 < l_3, l_5$. There is no way of labeling such that the above is satisfied. □

Lemma 3.18 excludes graphs with subgraphs of minimum paths that are combinations of more than one square with one triangle sharing a common face. However we can combine one square and one triangle with no problem.

Lemma 3.19 *The following graph with unit-cost links has an optimum LIRS.*

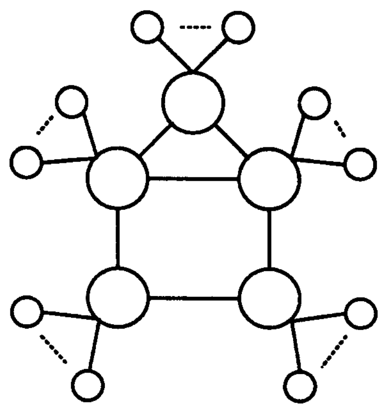


Figure 3.17

Lemma 3.20 *The bipartite graph* $\mathbf{K_{2,3}}$ *with unit-cost links has an optimum LIRS.*

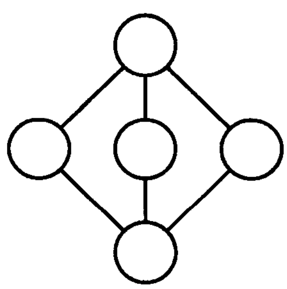


Figure 3.18

The proofs of the last two lemmas are straightforward and therefore omitted. It is not clear what the exact characterization is of the class of graphs with unit-cost links that have an optimum LIRS. Also the problem of the exact characterization of the class of graphs with unit-cost links that have a LIRS at all is open. We consider some interconnection networks that have an optimum LIRS.

Definition 3.21 *An* n-dimensional grid *of dimensions* $d_1, ..., d_n$ *is a network consisting of* $\Pi_{i=1}^{n} d_i$ *nodes* $(v_1, ..., v_n)$ *with* $0 \leq v_i < d$ *for every* $1 \leq i \leq n$*, where there is a link between node* $v = (v_1, ..., v_n)$ *and node* $w = (w_1, ..., w_n)$ *if and only if there is an* i*,* $1 \leq i \leq n$*, such that* $v_i = (w_i \pm 1)$ *and* $v_j = w_j$ *for all* $j \neq i$*.*

Lemma 3.22 *Any* n-dimensional grid *with unit-cost links has an optimum LIRS.*

Proof. The labels of the nodes form a natural linear ordering lexicographically, thus there is no need to relabel them. We label the links recursively as follows. For each chain $(v_1,...,v_{n-1},x_n)$ where x_n varies from 0 to $d_n - 1$ and the v_i's are fixed, we label the down-link $((v_1, ., v_{n-1}, x_n), (v_1, ., v_{n-1}, x_n - 1)$ by (0,0,...,0). The up-link $((v_1, ., v_{n-1}, x_n), (v_1, ., v_{n-1}, x_n + 1)$ is labeled by $(v_1, ., v_{n-1}, x_n + 1)$. Then within the chain, messages are routed optimally. Now consider the next dimension of chains $(v_1,...,x_{n-1},v_n)$. The down-link for each node is labeled with $(v_1, ..., 0, v_n)$ and the up-link with the label of the up-link node. Again, the labelings so far guarantee an optimal routing. Continuing the process for each dimension gives an optimum routing for the whole network. The obtained optimum LORS can easily be transformed in an optimum LIRS. □

We note that the Interval Routing scheme presented in [LT86] for grids is actually a LIRS, since no wrap around is used there. Furthermore, the above labeling gives a routing scheme that coincides with the routing scheme of the original interconnection grid network, normally provided by the hardware.

A *hypercube network* is the special case of the n-dimensional grid where $d_i = 2$ for all i.

Corollary 3.23 *A* hypercube network *with unit-cost links has an optimum LIRS.*

An *n-dimensional torus* is an interconnection network similar to the n-dimensional grid. The only difference is that wrap around is allowed. Thus in the above definition for grids we only need to change the formula $v_i = (w_i \pm 1)$ into the formula $v_i = (w_i \pm 1) \ mod \ d_i$.

Lemma 3.24 *The* n-dimensional torus *with unit-cost links has an optimum LIRS if and only if* $d_i \leq 4$ *for each* i*.*

Proof. If $d_i \geq 5$ for some i, then we have a cycle of length 5 or more as a subgraph of minimum paths and an optimum LIRS is not possible in that case (Lemma 3.3). If $d_i \leq 4$ for each i, we can modify the labeling for the n-dimensional grid to obtain an optimum LIRS for the n-dimensional torus. Again the labels of the nodes form a natural linear ordering lexicographically. We label the links recursively as follows. For each cycle $(v_1,...,v_{n-1},x_n)$ where x_n varies from 0 to $d_n - 1$ and the v_i's are fixed, we label the links as follows. We label the down-link $((v_1, ., v_{n-1}, x_n),$

$(v_1,.,v_{n-1},x_n-1))$ by (0,0,...,0) and the up-link $((v_1,.,v_{n-1},x_n),(v_1,.,v_{n-1},x_n+1))$ by $(v_1,.,v_{n-1},x_n+1)$, if x_n is not equal to 0 or (d_n-1). We label the down-link $((v_1,.,v_{n-1},x_n),(v_1,.,v_{n-1},(x_n-1) \bmod d_n)$ by $(v_1,.,v_{n-1},(x_n-1) \bmod d_n)$ the up-link $((v_1,.,v_{n-1},x_n),(v_1,.,v_{n-1},(x_n+1) \bmod d_n)$ by (0,0,...,0), if x_n is equal to 0 or (d_n-1). Then within the cycle, messages are routed optimally, because $d_n \leq 4$. Now consider the next dimension of cycles $(v_1,...,x_{n-1},v_n)$. The down-link for each node is labeled with $(v_1,...,0,v_n)$ and the up-link with the label of the up-link node, if x_{n-1} is not equal to 0 or $d_{n-1}-1$. We label the down-link $((v_1,.,x_{n-1},v_n),(v_1,.,(x_n-1) \bmod d_{n-1},v_n)$ by $(v_1,.,(x_{n-1}-1) \bmod d_{n-1},v_n)$ and the up-link $((v_1,.,v_{n-1},x_n),(v_1,.,(x_{n-1}+1) \bmod d_{n-1},v_n)$ by $(v_1,...,0,v_n)$, if x_n is equal to 0 or (d_n-1). Again, the labelings so far guarantee an optimal routing. Continuing the process for each dimension gives an optimum routing for the whole network. The obtained optimum LORS can easily be transformed in an optimum LIRS. □

4 A Hierarchy of Linear Interval Routing Schemes

In [LT87] the concept of a *multi-labeling* scheme was introduced for Interval Routing. The idea there is to allow more than one label per link. This considerably increases the power of the Interval Routing scheme and allows a larger class of networks to have an IRS with optimum routing.

Definition 4.1 *A routing scheme is called a k-*Linear Interval Routing scheme *(or* k-LIRS *for short), where $k \geq 1$, if it employs the Linear Interval routing technique and there are at most k labels per link.*

It is clear that any network that has a k-LIRS also has a $(k+1)$-LIRS, regardless of whether the links have unit costs, fixed costs or dynamic costs. Let k-$\mathcal{LIRS}$ be the set of networks that have an optimum k-LIRS. We now show that the hierarchy is strict for networks with dynamic link-costs.

Theorem 4.2 *For every $k \geq 1$, k-$\mathcal{LIRS}$ is a* strict *subset of $(k+1)$-$\mathcal{LIRS}$ for the case of networks with dynamic link-costs.*

Proof. Consider the bipartite graph $\mathbf{K_{2,2k+1}}$. There is a partition of $\mathbf{K_{2,2k+1}}$ into say, V_1 and V_2 with $|V_1| = 2$ and $|V_2| = (2k+1)$ such that the edges of $\mathbf{K_{2,2k+1}}$ connect between nodes of V_1 ana V_2 only. Label the nodes in V_1 as 2 and 4, and the nodes in V_2 as $1,3,5,6,...,2k+3$. Consider node 2. Depending on the link-costs messages for nodes that are element of V_2 either can be routed via node 4 or can be routed directly. The worst case appears if all messages that are routed via node 4 are routed via one and the same link l incident to node 2. Let V' be the set of node-labels that must be covered by l. It is possible that l must cover parts of the interval $[5,2k+3]$. Let $v \in (5,2k+3]$ such that $v \notin V'$, i.e., if messages from node 2 to v are routed directly, then v must be deleted from $[5,2k+3]$. This causes a "gap" in the interval, i.e., $[5,2k+3]$ is divided into two subintervals. As a result it may happen that in order to cover V', l must be labeled with at least two intervals. The maximum number of gaps that can be obtained by deleting elements from the interval $[5,2k+3]$ is $(k-1)$. Thus l needs at most k labels to cover the nodes in $V' \cap [5,2k+3]$. It is straightforward to verify that in order to cover the other nodes at most one extra label is needed. Similarly it can be shown for the other links of $\mathbf{K_{2,2k+1}}$ that at most $(k+1)$ labels are needed. Hence $\mathbf{K_{2,2k+1}} \in (k+1)-\mathcal{LIRS}$.

Suppose now that the nodes of $\mathbf{K_{2,2k+1}}$ are labeled somehow. Let the nodes in V_2 be labeled with, say, $x_1, ..., x_{2k+1}$, in order. If the labels of both of the nodes in V_1, say y_1 and y_2, are greater than x_1, then by assigning the appropriate cost to each link we can create k gaps with the sequence $x_2, x_4, x_6, ..., y_i, ..., x_{2k}$, where $i = 1$ or 2. Thus link (x_1, y_i) must be labeled with at least $(k+1)$ labels. Similarly, if $y_i < x_1$ for some i, we can create k gaps in the sequence $y_i, x_3, x_5, x_7, ..., x_{2k+1}$. It follows that link (x_1, y_i) must be labeled with at least $(k+1)$ labels. Thus there is no way that $\mathbf{K_{2,2k+1}}$ can be labeled with only k labels. □

In [BLT90] it is shown that k-$\mathcal{IRS}$ is contained in $(k+1)$-$\mathcal{PRS}$ for every $k > 1$. This is actually trivially true for LIRS also.

Theorem 4.3 *For $k > 1$, k-$\mathcal{IRS} \subset (k+1)$-$\mathcal{LIRS}$.*

Proof. Every time there is a wrap around in a k-IRS, add an extra label of 1 to the corresponding link in k-LIRS to obtain at most a $(k+1)$-LIRS. □

The characterization of k-$\mathcal{LIRS}$ for $k > 1$ remains an open question.

5 Acknowledgements

We thank Hans Bodlaender and Gerard Tel for useful comments and suggestions for improvements.

References

[AGR89] Y. Afek, E. Gafni, M. Ricklin. *Upper and Lower Bounds for Routing Schemes in Dynamic Networks.* Proc. of the 30th Annual Symp. on Foundations of Computer Science, 1989, pp. 370-375.

[BLT90] E.M. Bakker, J. van Leeuwen, R.B. Tan. *Prefix Routing Schemes in Dynamic Networks.* Techn. Rep. RUU-CS-90-10, Dept. of Computer Science, Utrecht University, Utrecht, March 1990.

[BG87] D.P. Bertsekas, R.G. Gallager. *Data Networks.* Prentice-Hall Inc., Englewood Cliffs, N.J., 1987.

[FJ86] G.N. Frederickson, R. Janardan. *Separator-Based Strategies for Efficient Message Routing.* Proc. 27th IEEE Annual Symp. on Foundations of Computer Science, 1986, pp. 428-437.

[FJ88] G.N. Frederickson, R. Janardan. *Designing Networks with Compact Routing Tables.* Algorithmica, Vol. 3, 1988, pp. 171-190.

[LT86] J. van Leeuwen, R.B. Tan. *Computer Networks with Compact Routing Tables.* In, G. Rozenberg and A. Salomaa (Eds.), The Book of L, Springer-Verlag, Berlin, 1986, pp. 259-273.

[LT87] J. van Leeuwen, R.B. Tan. *Interval Routing.* The Computer Journal, Vol. 30, No. 4, 1987, pp. 298-307.

[SK85] N. Santoro, R. Khatib. *Labelling and Implicit Routing in Networks.* The Computer Journal, Vol. 28, No. 1, February 1985, pp. 5-8.